

ACCURATE AND DISCERNIBLE PHOTOCOLLAGES

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Jordan William Miller

©Jordan William Miller, January 2010. All rights reserved.

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

ABSTRACT

There currently exist several techniques for selecting and combining images from a digital image library into a single image so that the result meets certain prespecified visual criteria. Image mosaic methods, first explored by Connors and Trivedi[18], arrange library images according to some tiling arrangement, often a regular grid, so that the combination of images, when viewed as a whole, resembles some input target image. Other techniques, such as Autocollage of Rother et al.[78], seek only to combine images in an interesting and visually pleasing manner, according to certain composition principles, without attempting to approximate any target image. Each of these techniques provide a myriad of creative options for artists who wish to combine several levels of meaning into a single image or who wish to exploit the meaning and symbolism contained in each of a large set of images through an efficient and easy process.

We first examine the most notable and successful of these methods, and summarize the advantages and limitations of each. We then formulate a set of goals for an image collage system that combines the advantages of these methods while addressing and mitigating the drawbacks. Particularly, we propose a system for creating photocollages that approximate a target image as an aggregation of smaller images, chosen from a large library, so that interesting visual correspondences between images are exploited. In this way, we allow users to create collages in which multiple layers of meaning are encoded, with meaningful visual links between each layer. In service of this goal, we ensure that the images used are as large as possible and are combined in such a way that boundaries between images are not immediately apparent, as in Autocollage. This has required us to apply a multiscale approach to searching and comparing images from a large database, which achieves both speed and accuracy. We also propose a new framework for color post-processing, and propose novel techniques for decomposing images according to object and texture information.

ACKNOWLEDGEMENTS

I wish to express the most sincere gratitude to my supervisor, Dr. David Mould, for his inspiration, guidance, and support. Also, I thank the members of the committee for contributing valuable input and comments.

We have made good use of the Edge Detection and Image Segmentation (EDISON) System [38, 63, 17, 16] provided by the Robust Image Understanding Laboratory at Rutgers, and we thank them for allowing its use.

For the large library of images required to test our results, we used photographs from the Library of Congress Flickr photoset [66]. We thank the Library of Congress for making these images available.

This thesis is dedicated to Mary Jean Hande.

CONTENTS

Permission to Use	i
Abstract	ii
Acknowledgements	iii
Contents	v
List of Figures	vii
1 Introduction	1
1.1 Accuracy Vs. Discernibility	1
1.2 Objectives	6
1.3 Overview of Method	8
1.4 Novel Contributions	9
1.5 Organization of Thesis	11
2 Literature Review and Related Works	12
2.1 Preamble	12
2.2 Related Work	13
2.2.1 Art and Illustration	13
2.2.2 Image Mosaics	15
2.2.3 Collages	21
2.3 Literature Review	24
2.3.1 Mosaics	24
2.3.2 Image Abstraction and Non-photorealistic Rendering	25
2.3.3 Texture Synthesis	28
2.3.4 Image Completion	30
2.3.5 Frequency Space Image Synthesis	32
2.4 Summary	32
3 Texture Amplitude and Saliency Measures	34
3.1 Overview	34
3.2 Requirements for a Texture Contrast Map	34
3.3 Properties of Texture Images	35
3.4 A Basic Texture Contrast Map	36
3.5 A Better Texture Contrast Map	36
3.5.1 Histogram Folding	37
3.5.2 Evaluation	39
3.6 Saliency Maps	42
3.7 Edge Maps	43
4 Matching	47
4.1 Introduction	47
4.1.1 Achieving Accuracy and Discernibility	47
4.2 Collage Tiles	49
4.3 Matching Problem Formalization	53
4.3.1 Calculation of Energy Terms	54
4.3.2 Details of Energy Term Calculations	55
4.4 Acceleration of the Matching Process	59

4.4.1	Hierarchy	61
4.5	Refinement	64
5	Compositing	67
5.1	Overview	67
5.2	Seam Repair	67
5.2.1	Path Cost Voronoi Diagrams	68
6	Post-Processing	72
6.1	Motivation	72
6.2	Our Approach	73
6.2.1	Hybrid Images	74
6.3	Method	75
6.3.1	Detail Enhancement	79
6.3.2	Texture Simplification	81
7	Results and Evaluation	85
7.1	Overview of the Algorithm	85
7.2	Discussion of Results	89
7.2.1	Selection of Target Image	90
7.2.2	Accuracy	91
7.2.3	Discernibility	91
7.2.4	Compositing	92
7.2.5	Image Library	92
7.2.6	Tile Image Scale	93
7.2.7	Refinement and Scaling	95
7.2.8	Edge Weights in Matching and Refinement	97
7.2.9	Corner and Salience Maps in Refinement	97
7.2.10	Luminance and Chrominance	98
7.2.11	Texture Contrast	98
7.2.12	Library Reduction	100
7.2.13	Scale Hierarchy	100
7.2.14	Postprocessing	101
7.3	Efficiency and Performance	102
7.4	Failures and Limitations	103
7.5	Selected Results	106
8	Conclusion and Future Work	114
8.1	Conclusion	114
8.2	Summary of Future Work	116
	References	118

LIST OF FIGURES

1.1	“All is Vanity” [84] by Charles E. Gilbert.	2
1.2	1.2a “Calvary” [84] by Octavio Ocampo.	3
1.3	“Water” [84] by Giuseppe Arcimboldo.	4
1.4	1.4a “Marlene” [84] by Octavio Ocampo.	5
1.5	“NICHOLAS (Little League Pitcher)” [53] by Ken Knowlton.	6
1.6	The process of iterative refinement.	10
2.1	“The Forgetting Room”, by Nick Bantock [4].	13
2.2	Detail from the graphic novel <i>Arkham Asylum: A Serious House on Serious Earth</i> , illustrated by Dave McKean [64].	14
2.3	The portrait “Vertumnus” by Arcimboldo [21].	15
2.4	“Gala contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln (Homage to Rothko)” by Salvador Dali [20].	16
2.5	An example of Finkelstein and Range’s Image Mosaic [36].	17
2.6	A cut-out image mosaic from the paper by Orchard and Kaplan [69].	18
2.7	A Jigsaw Image Mosaic [51].	20
2.8	An example of a collage created by Autocollage [78].	22
2.9	The results of various collage and mosaic rendering algorithms.	23
2.10	A mosaic by Hausner [43] exhibiting tiles of size and orientation adapted from the target image.	25
2.11	2.11a The original image. 2.11b The abstracted image via the method of DeCarlo and Santella [22].	27
2.12	4.2a The original image. 2.12b The abstracted image via the method of Orzan et al. [70].	27
2.13	2.13a The original image. 2.13b A painterly rendering via Hertzmann’s method [46] with a coarse brush.	28
2.14	2.14a The target image. 2.14b The exemplar texture, to be transfered to the target image.	30
2.15	A demonstration of Hays and Efros’s method [44] for scene completion.	31
2.16	A hybrid image, by the method of Oliva and Torralba [68].	32
3.1	3.1a Texture image I . 3.1b $ H_m = I - M(I) $ where M represents the median filter for some radius. 3.1c $M(H_m)$	35
3.2	3.2a Image I . 3.2b $ H_m = I - M(I, r) $. 3.2c $M(H_m , r)$	37
3.3	One iteration of the histogram folding process, which occurs during one iteration of operation k	38
3.4	3.4a–3.4c A view of each term in the sum of images used to compose the final, spatially precise texture contrast map.	39
3.5	Two images (3.5a and 3.5c and their texture contrast maps	40
3.6	A sample of Brodatz textures [12] that do and do not agree with our assumptions.	41
3.7	3.7a Original image. 3.7b Texture contrast by Lozano1 and Escolano [59].	42
3.8	3.8a Original image. 3.8b Texture contrast by Kokkinos et al. [54].	43
3.9	3.9a Original image. 3.9b Texture contrast by Carson et al. [14].	44
3.10	3.10a Original image $I_0 = I$. 3.10b One iteration of corner map $I_1 = I - M(I_0) $	45
3.11	An improved image edge map.	46
4.1	4.1a A small portion of library image, composited into a square tile.	48
4.2	“The Great Paranoiac” [84] by Salvatore Dali.	49
4.3	A detailed view of an image mosaic produced by Orchard and Kaplan [69].	50

4.4	Detail of the collage “Brooke”, by Derek Gores [40]. For this portion of the collage, it appears that the artist has segmented the target image by color.	50
4.5	4.5a Mean-shift segmentation, with hand-chosen parameters, via EDISON [38]. . . .	52
4.6	Three mean shift segmentations, produced by EDISON [38] with three sets of parameters. Note that for all segmentations the tiles vary widely in size, shape, and position.	52
4.7	4.7a Feature/edge map. 4.7b Voronoi tiling, with centroids placed randomly, but distributed according to the feature map.	53
4.8	There is a high amount of noise in the image mosaic, caused not only by tile boundaries, but by high-frequency structure in the tile images.	56
4.9	4.9a Region of the target image containing a high-contrast step edge.	57
4.10	4.10a Target image. 4.10c Composition with low s_{ROI}	60
4.11	For each tile, outlined here in green, each pixel of the library image,	61
4.12	A visual guide to the matching process.	63
4.13	4.13a Collage. 4.13b Original target image. 4.13c Error between collage and original.	64
4.14	4.14a Target Image. 4.14a The edge map F , for increasing discernibility around edges.	66
5.1	Three intermediate stages in the Path Cost Voronoi diagram calculation.	69
5.2	5.2a Original image I to be composited. 5.2b Edge map e of I	70
5.3	5.3a Composite with a narrow maximum feather radius.	71
6.1	6.1a The mosaic of Finkelstein and Range [36] has been color-corrected	73
6.2	6.2a The hue shifting scheme of Orchard and Kaplan sometimes produces odd-looking results	74
6.4	A modified hybrid image, with base layer created from a cross bilateral filter	76
6.5	A hybrid image, using a method similar to the that presented in Hybrid Images [68]. . . .	77
6.6	A comparison of two double image methods.	78
6.8	6.8a The original image. 6.8b The texture-simplified collage image.	79
6.10	6.10a The target image T . 6.10b The color corrected collage.	81
6.11	6.11a Original Image. 6.11b Image abstraction via the method of Orzan et al. [70]. . . .	84
7.1	A poorly produced collage and the corresponding target image.	90
7.2	7.2a Target image. 7.2b Photocollage. 7.2c Absolute error between target image and collage. Note that overall, the error is low.	92
7.3	7.3a Original library image. 7.3b Image in the composition. The face has not been included, but the shape remains as part of the tile boundary.	93
7.4	Two examples of successful compositions via path cost Voronoi tessellations.	93
7.5	7.5a Our composition method. 7.5b An image produced by Microsoft Autocollage 2008 [79].	94
7.6	An illustration of the effect of library size on collage quality.	95
7.7	An illustration of the effect of image-size to tile-size ratio on <i>discernibility</i> and <i>accuracy</i>	96
7.8	7.8a Original target image. 7.8b Collage, without edge weights.	97
7.9	The effect of changes to luminance/chrominance balance.	99
7.10	The effect of changes to the balance of texture contrast in the matching calculation. . . .	99
7.11	textbf7.11a Detail of photocollage, before post processing.	101
7.12	7.12a Target image. 7.12b Collage, before postprocessing.	102
7.13	A collage emphasizing discernibility, made from 685 images. Individual collage images are circled.	106
7.14	A collage emphasizing accuracy, made from 685 images. Individual collage images are identified.	107
7.15	A collage emphasizing discernibility. The size of the library is 837 images, and two refinement passes are used.	108
7.16	A collage emphasizing accuracy. The size of the library is 837 images, and one refinement pass is used.	109

7.17	A collage made from 30 images of the author, with one refinement pass. Additional post processing was done to emphasize edges in the library images.	110
7.18	A collage emphasizing discernibility. 837 images are used, and one refinement pass.	111
7.19	A collage emphasizing discernibility, made from 685 images. Saturation has been enhanced to improve discernibility.	112
7.20	A collage made from 685 images, with two refinement passes. 7.20b The result of Orchard and Kaplan [69] is shown for comparison.	113

CHAPTER 1

INTRODUCTION

Image mosaics, double images, and collages are various visual artistic forms that order images in relation to each other and imply meaning beyond what is suggested by the images individually. Double images, which depict two images simultaneously through the sharing of image objects for dual purposes, have been created by artists for centuries. Image mosaics, while appearing in many limited forms in western art, have become especially viable in the past fifteen years due to computational methods for creating them. The image mosaic is a mosaic rendered such that each tile, rather than containing a solid color, contains another smaller image. It is an art form that uses images, arranged just so, to render other images.

When we create art, we render the world from something formless and confusing into something manageable and understandable. Through his or her art, the visual artist creates a narrative, assigning meaning to the meaningless, and orders the disparate and contradictory elements of nature into a logical structure of causes and effects. Through the direct comparison and association of images, a visual syntax is established and a text is created that is richer than the sum of its elements. By connecting ideas, new ideas are created. The meanings associated with the elements and their various relations may be enforced as a static property of the text or dynamically created through the interpretation of each viewer. In double images, image mosaics, and photocollages (which we will describe in this thesis) these connections and associations are made explicit.

We will begin this chapter with a discussion the goals and ideals of the image mosaic process and qualities of successful image mosaics. We will discuss properties of *double images* and techniques employed by artists to code multiple images into one. We will then compare these techniques with each other and distill these ideas into a set of achievable properties and policies for creating a new type of double image. Through this, we will identify the differences between the image mosaics and the photocollage, which we introduce in this thesis.

1.1 Accuracy Vs. Discernibility

Orchard and Kaplan, in their paper “Cut-Out Image Mosaics” [69], refer to the qualities of *Accuracy* and *Discernibility* as the main goals of the image mosaic process. They define *accuracy* as the quality

of presenting an accurate depiction of the target image. The target image is that which they wish to represent as an arrangement of mosaic tiles. *Discernibility* is then, as they put it, the “legibility” of the images within the tiles. It is easy to trade off between accuracy and discernibility in a mosaic simply by adjusting the size of the mosaic tiles; for example, one may produce a perfectly accurate mosaic by allowing the size of each tile image to be reduced to a single pixel. On the other hand, a “mosaic” of one tile, which takes up the entire image, may have perfect discernibility without any accuracy at all.



(a)

Figure 1.1: “All is Vanity” [84] by Charles E. Gilbert. The single “tile image” of a lady looking into a mirror is perfectly discernible, but the “target image” of a skull is also very accurately depicted.

However, it is possible to find cases in which this “single tile mosaic” is actually successful in achieving both accuracy and discernibility; in these cases, the successful result is obviously really not a mosaic at all but something that is called, in Seckel’s book [84], a *double image*. These double images have taken many forms, including the *anthropomorphic landscape* paintings and engravings that were popular after the sixteenth century, produced by artists such as Matthaus Merian [84]. The images may be thought of as the result of a sort of object-wise matching process; in Charles E. Gilbert’s “All is Vanity” [84] (see figure 1.1), the head of the lady, in and out of the reflection of the mirror, has been matched with the eyes of the skull, and the mirror itself has been matched with the crown of the skull.

In these double images, there is a clear distinction between the “primary” and “secondary” images. The secondary image is what we will refer to, in the context of image mosaics, as the target image, or the image that we wish to depict secondarily through the manipulation of some other image or set of images. The primary image (or images) is, then, the image that we use to

depict the secondary image through some creative manipulations. In the case of image mosaics, the secondary image is depicted through the combination of many primary images. Perhaps, using this terminology, we may speak of *accuracy* as fidelity to the secondary image and *discernibility* as fidelity to the primary image. In any case, the details of the double image mostly belong to the primary image; that is, an initial inspection of the details in a double image will usually reveal the primary image, whereas further inspection may reveal the secondary image.

Even so, it is difficult to assign responsibilities for semantic representation of objects between the primary and secondary images. One strategy is to take a *base/detail* approach in which we say that the details of the double image belong to the primary image while the base structures belong to the secondary image. For example, viewing a painting by the 16th century Italian portraitist Arcimboldo (1.3) with one’s nose close to the page will reveal only the fruits and grains of the primary image, while taking a step back will reveal the base structure of the human face. See figure 1.2 for an example of a double image by Ocampo that we have crudely divided into base and detail layers. This base/detail dichotomy is, of course, not strictly observed, and this accounts for much of the magic and cleverness of double images. The *Hybrid Images*, produced by Oliva et al. [68], represent an extreme case of this dichotomy; only the high spatial frequency data of the primary image is used with the low spatial frequency data of only the secondary image. We will discuss these ideas, as they relate to our implementation, in more detail in chapter 6.

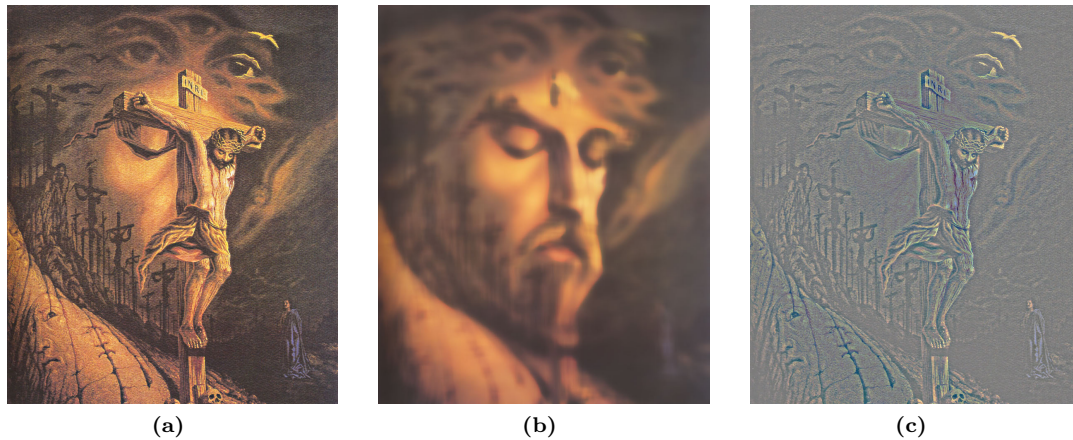


Figure 1.2: **1.2a** “Calvary” [84] by Octavio Ocampo. The primary image depicts the crucified Christ, while the secondary image depicts Christ’s face. The cross and outstretched arms in the primary image form the eyes, nose, and mouth in the secondary image. One of the eyes in the secondary image has been matched with the head in the primary image. Note that there are several smaller secondary images hidden throughout the painting. **1.2b** By taking the median filter (which has an edge-preserving quality) of the image, we are able to produce a base layer in which most of the influence from the primary image has been removed. The secondary image, depicting the face of Christ, is then clearly visible. **1.2c** The detail layer, obtained by subtracting the base layer from the original image. The contrast has been enhanced for visibility. The influence of the secondary image is minimal, while the primary image is clearly discernible.

The object-wise quality of matching is more pronounced in the double images of some artists than others, such as Arcimboldo who depicted the subjects of his portraits as aggregations of fruits, animals, and other objects. Figure 1.3 shows an Arcimboldo portrait using sea creatures as the matching primitives. In images such as these, the objects of the primary image, which may be thought of as the matching primitives, are aggregated independently of any high-level context at the object level and independent of any aim other than to represent the target image. These images may be said to have a somewhat more “mosaic” quality, although the matching primitives do interact with each other through overlapping, shadowing, reflection and occlusion, establishing a minimal level of inter-object semantic coherence. This is in contrast to the work of the anthropomorphic landscape artists and others who attempt to maintain a valid semantic context for the image objects, in which the placement of objects with respect to each other has some discernible meaning, apart from their function as matching primitives for the target image.



(a)

Figure 1.3: “Water” [84] by Giuseppe Arcimboldo.

This approach to object-wise matching has been taken up in computer graphics in the *3D Collage* [37] and in two dimensions with the *Jigsaw Image Mosaic* [52] and *Puzzle Image Mosaic* [9]. The latter of these may be categorized, with reasonable confidence, as a mosaic form, since interaction between matching primitives, and thus an overall object level context, has been completely eliminated. Moving into this territory of mosaics, we see that the balance between accuracy and discernibility becomes difficult to strike, at least in automatically generated mosaics. Certainly, the production of double images, particularly image mosaics, is somewhat algorithmic and thus amenable to computer-aided automation.

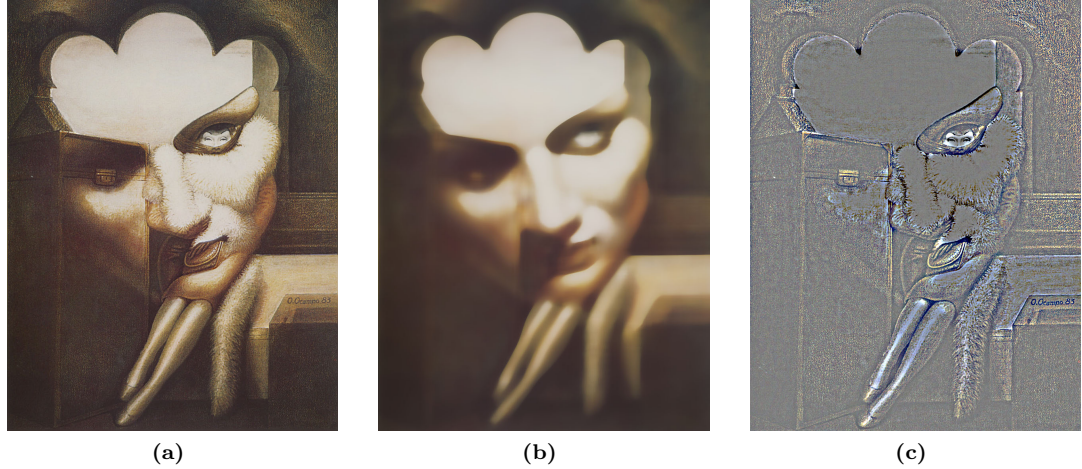


Figure 1.4: **1.4a** “Marlene” [84] by Octavio Ocampo. The primary image depicts an aristocratic woman sitting near a trunk, while the secondary image depicts a woman’s face. As in “Calvary” (and many other double images of this type), the head in the primary image is matched with one of the eyes in the secondary image. **1.4b** The base layer, in which the secondary image is more prominent. **1.4c** The detail layer, in which the primary image is more dominant (the contrast has been enhanced for this figure). Since the artist has used shape, as well as shading, to suggest the secondary image, we can see some of the secondary image even in the isolated primary image.

Pareidolia Interestingly, in many (or perhaps even in most) double images the secondary image depicts a human face as in anthropomorphic landscapes, all of Arcimboldo’s portraits, and many of the most striking double images by Dali, Del-Prete, Knowlton, Muniz, Ocampo, and Orosz [84]. The phenomenon of observing seemingly meaningful patterns and structures in data with unrelated meaning, or no meaning at all, is known as Pareidolia [41]. It is known that the human visual system is specially adapted to recognize human faces, even from images that semantically are not at all related [41]. Therefore, it is conceivable that less accuracy is strictly required in depicting, through a double image, the secondary image of a face, since the human visual system will automatically make up for some of the deficiencies. (Although trivial, it is interesting that the matching of a human head in the primary image with an eye in the secondary image is a common technique.)

Hybrid Images In hybrid-images, edges from the secondary image are blurred out and thus may only be resolved at a far viewing distance. Double images, on the other hand, have the quality that both secondary and primary images are observable at close viewing. This may be due to the preservation of hard edges from the secondary image in the double image; edge data from the secondary image is allowed to be propagated through all scales (edges of this type are called “old edges” by Orzan et al. [70]).

At this point it is necessary to draw attention to the distinction in the use of shading and shape to convey the secondary image of a double image. Classic mosaics rely mostly on the shading within



(a)

Figure 1.5: “NICHOLAS (Little League Pitcher)” [53] by Ken Knowlton, a true photo mosaic. The representation of the target image (a young ball player) is achieved both through the mosaic layout of the atomic elements, and the content of the elements themselves.

the matching primitives to convey the content of the secondary image. The matching primitives contain no shapes to assist in the conveyance of the target image. Double images that use more sophisticated correspondences and fewer matching primitives with richer context rely on shape as well as shading to convey the content of the secondary image, as in figure 1.4. The difference between the characteristics of *shape* and *shading* is that shape relies on hard, well defined edges, whereas shading relies only on large-scale, softly varying features. In some sense, hybrid images hide the secondary image in the shading. Other double images do this as well but also rely on the shapes of the objects depicted therein to hold meaning in both the primary and secondary images. For example, in figure 1.4, the edge of the woman’s arm in the primary image is also the edge of the jaw in the secondary image.

So, it is reasonable to propose the creation of a new type of hybrid image from edge-preserving decompositions of images rather than purely spectral decompositions, as in Oliva and Torralba’s work. However, we must also ensure that the edges preserved are the *same* edges in both the primary and secondary images. This may be done using the cross-bilateral filter, in manner similar to the image denoising and detail transfer work of Petschnigg et al. [74] and Eisemann and Durand [31]; we will discuss this in more depth in chapter 6.

1.2 Objectives

The term *image mosaic* seems to have been used to describe any double image in which the aggregation of primary images or objects are each arranged and rendered disjointly from one another. In this case, the target image is partitioned into a set of tiles, and each tile is treated independently of

the others, except that each tile is matched to a different region of the same target image. Figure 1.5 shows an image mosaic by Knowlton, depicting the face of a young baseball fan composed from baseball cards.

In “Autocollage”, a collaging system by Rother et al. [78], there is no secondary image; the products of Autocollage are not double images. However, there are some global aesthetic constraints for which tiles must be considered in relation to each other. In this way, a global context is created. Furthermore, during rendering, image object boundaries are consulted in the establishment of final tile boundaries. In distinguishing collages from mosaics, we have decided that for collages the tile partition is not strict; overlap between objects may occur where it is appropriate, and there exists, if only in a very basic form, some context that encompasses all of the matching primitives. Perhaps there need not be any physical, three dimensional space as there is in many double images, in which the matching primitives exist; however, object-object interaction should occur through the establishment of boundaries between tile image regions, which are drawn around image objects. This context exists independently of the target image matching constraint, or secondary image constraint. In traditional art, collage artists have used a diverse array of techniques [48]; certainly, the form of collage has allowed for irregular compositions involving the irregular placement of irregularly shaped objects. These collage characteristics may be observed in the works of such artists as Nick Bantock and Dave McKean (see chapter 2).

We have discussed the case of the double image, in which one single image accurately expresses some other image while maintaining its own discernibility. While it may be difficult to automatically construct such a clever image (although we have made progress toward a solution, as discussed in chapter 6), we may partition our target image into smaller sub-images and tackle several easier versions of the problem instead. This alone is the general approach of image mosaic methods, but we have more ambitious goals for our project and set out a list of more specific objectives.

Images should be placed in tiles so that there are interesting correspondences between the tile images and the target image—not merely correspondences between average color or color gradient but also between salient image structures. Previous methods [69, 10, 99] have done this, but only with small-scale, high-frequency image features. We assume that the more of the target image we can match to a single collage image, the more interesting the results; thus, we attempt to match large tiles to large images. Again, there have been previous attempts at this goal [69, 10], but the results still rely heavily on extensive subdivision around features of the target image to achieve accuracy. We should not have to rely on the aggregation of a large number of matching primitives of correct mean color. The tile images should interact with each other within some basic context, and so we would like tiles to blend into each other seamlessly, but also we would like for tile boundaries to respect image object and structure boundaries; ideally, the viewer should have to examine the image closely to determine where the tile boundaries are. As always, we wish for the contents of

each tile to be as discernible as possible but also for the whole collage to depict the target image as accurately as possible.

1.3 Overview of Method

In this section, we will provide a brief outline of the entire process of creating a photocollage. In subsequent chapters, we will expand each of the following steps and provide full descriptions of the methods used. The production of an image-collage is performed in six main steps:

- preprocessing of the target image to obtain edge maps and image feature maps;
- partition of target image into sets of tiles, by Voronoi tessellation or some other segmentation method; The photocollage process works by iterative refinement, and for each level of refinement we need a complete set of tiles;
- for each refinement level:
 - matching of each tile with a fragment of some image from the library;
 - composition of tile images to create the final collage, including tile boundary adjustments;
 - color correction;
 - final error accounting and selection of tiles to be reprocessed in the next refinement level.

Preprocessing We extract from the target image some supplementary feature information to inform the collaging process. Particularly, we determine which parts of the image may require tighter error thresholds and which parts can tolerate looser matches. We also extract texture information, and produce downsampled versions of the image for faster matching. Additionally, each image in the image library from which tile images will be drawn is preprocessed and downsampled. The results are saved to special files so that a given library need only be processed once.

Partitioning We start by producing a collage for a very coarse, large-tiled partition of the target image and then replace poorly matched regions with smaller, presumably better-matched tiles in the next refinement stage, while leaving well matched regions as they are. The number of refinement stages required for a sufficiently accurate collage varies according to the target image being processed, the coarseness of the initial partition, the difference in coarseness between each refinement level, and the size of the image library. We have achieved good results using an initially very coarse partition (approximately ten tiles for an image of size 640×480) with the next refinement level having between four and six tiles for every one in the previous level and two refinement levels after the first. Please see figure 1.6 and chapter 7 for a demonstration of results obtained with these parameters. Chapter 4 gives a more detailed description of the tile partition process.

Matching For each tile, we must search the library for an appropriate match. First, we must determine what proportion of each tile image we would like to use; then for each image in the library we must compare every possible translation with respect to the region of the target image within the tile. This comparison process itself consists of several stages. In the first stage, we do not attempt to find the optimal translation but only to find a subset of images from the library that are likely to contain good matches. In subsequent stages, we compare every translation of the library image against the tile and further pare down the subset of legitimate image candidates. In each of these stages, we increase the resolution from the previous stage so that matching is more accurate, even as we search a smaller library than the previous stage. We provide a complete description of this process in chapter 4.

Compositing Once we have selected a single image for each tile along with an appropriate scale and spatial translation, we must render the collage. We must also update the boundaries of our tiles, since they were initially chosen without any knowledge of what images would be contained therein. Each tile is slightly overmatched outside its boundaries so that tile boundaries may be extended where needed without degrading the quality of the result. Boundaries are adjusted to respect image objects and structures within the tile images. A color correction operation is then performed to nudge the collage image a little closer to the target image without the appearance of “cheating”. See chapter 5 for a more detailed description of the compositing process.

Error Accounting Since the collage produced at each refinement level is not entirely accurate, we must calculate an error map between the collage and the target image. In some regions, there will be significant error where neither a good library image match could be found, nor could color correction make up for the deficit. For each tile in the partition of the next refinement level, we calculate an error density, which is the sum of the errors within the tile divided by the tile area. Tiles which have an error density above a certain threshold are “activated” for processing in the next refinement level. Tiles whose error density is below this threshold are kept just as they are, since over-refinement is not desirable.

This set of steps is repeated, with a finer partition, until a desirable collage is achieved.

1.4 Novel Contributions

In this thesis we introduce several novel techniques and approaches for solving a variety of image processing and computer graphics problems. The following is a list of the novel contributions and a brief description of each.

- We gather together general techniques for scale-based analysis and iterative refinement from the NPR (non-photorealistic rendering) literature and apply them to the problem of photo-

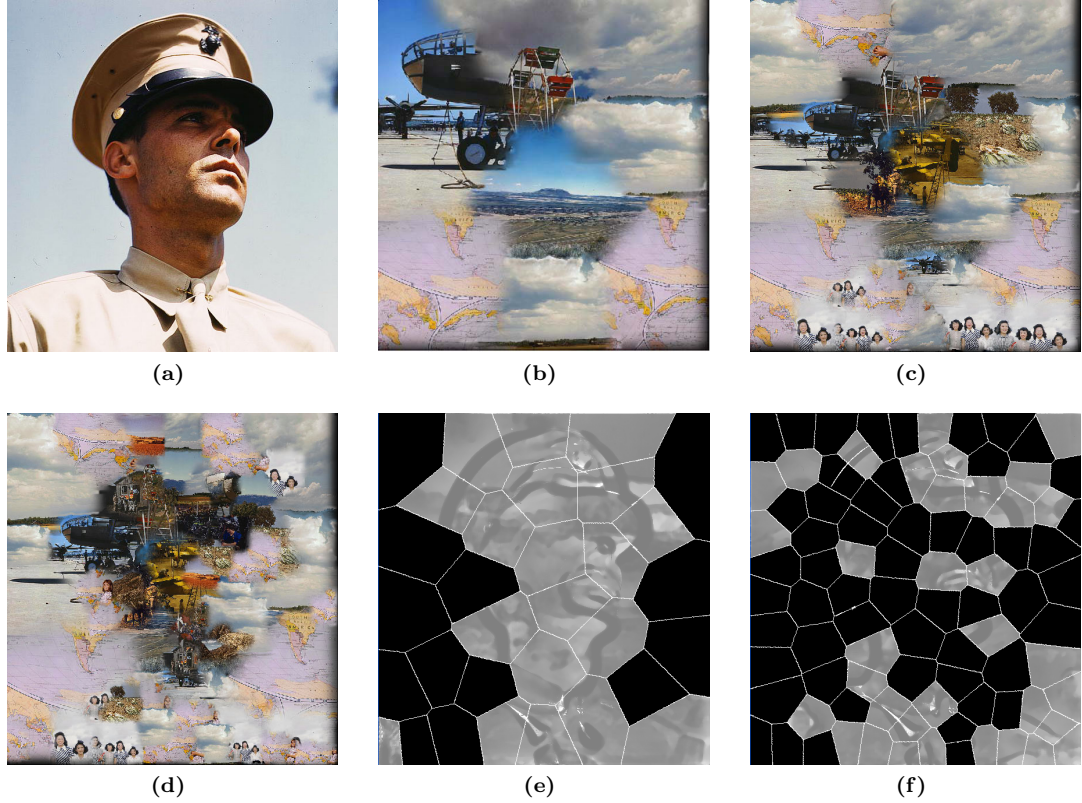


Figure 1.6: The process of iterative refinement. **1.6a** Target image. **1.6b** First pass. **1.6c** Second pass. **1.6d** Third and final pass. **1.6e**, **1.6f** The active tiles for each next refinement level, overlaid on the error from the previous level.

collages. Particularly, we frame the ideas of multi-scale image processing in terms of *accuracy* and *discernibility*, as Orchard and Kaplan defined these terms (chapter 1 and 4).

- While spectral decompositions and low dimensionality signatures have been used to accelerate the process of image database querying and matching in image mosaics, we propose the use of a multiscale approach, employing successive searches in increasing detail (chapter 4).
- We introduce a new measurement of texture contrast and simultaneously a new measurement of image salience, both of which are useful for image abstraction, texture simplification, and other image processing objectives (chapter 3).
- We introduce the use of Path Cost Voronoi diagrams as a simple and fast content-sensitive image segmentation scheme for the purpose of composing the collage image set as a whole (chapter 5).
- While a few methods exist for hiding one image in another, we propose another method, motivated by double images in art and recent work in edge-preserving image decompositions, that has wider applicability (chapter 6).

1.5 Organization of Thesis

Before we begin building on the basis we have laid out for a photocollage system, in chapter 2 we will first review the literature that has been produced thus far on this and related subjects. We will further look into some artistic traditions and derive motivation from them for our own artistic intentions. Once we have established the place of this work within the context of the work that has come before it, we will proceed to outline the development of the system in a process-wise fashion, starting with the early processes and moving to the later ones. In chapter 3, we describe the techniques for obtaining the edge and texture information that will be used throughout the entire photocollage process. In chapter 4, we will discuss our strategy for decomposing the target image into tiles and matching each of these tiles with an image from a library. Through this discussion, we will explain how each small process fits into the satisfaction of the goals we have established in this chapter. Then, in chapter 5, we will move to the next stage, which is composition of the selected images into a collage and the image processing technology that is used to ensure an attractive result that adheres to our artistic philosophy. Following in chapter 6 is a description of a postprocessing scheme for strengthening perceptual fidelity to the target image. In chapter 7, we review the products of this work, describing the advantages and disadvantages of each component and the work as a whole. Finally, in chapter 8, we provide our final thoughts on the system and its place within the context of other non-photorealistic rendering tools and make suggestions for future work.

CHAPTER 2

LITERATURE REVIEW AND RELATED WORKS

2.1 Preamble

The problem of the image collage has required us to look for inspiration from several different areas of computer graphics research. Various different types of image collage frameworks already exist, each of which has its own advantages and drawbacks. In addition to this, systems have been developed for texture synthesis, artistic rendering, image abstraction and other non-photorealistic rendering applications that are informative to the problem we are examining. Following is a brief discussion of the main developments in each of these areas that have informed and motivated this project.

We begin by discussing significant artists and works of art that have, from a creative and aesthetic point of view, inspired this work; particularly, we discuss the contributions of Nick Bantock and Dave McKean and the aspects of their works we are interested in studying and reproducing.

We then review the most significant related work in the area of image mosaics and collages. We will discuss various different rendering problems within the sphere of the collage and mosaic forms and the details of the techniques, developed throughout the past fifteen years, for solving these problems and creating creatively and visually satisfying image mosaics and collages. We will look closely at the most significant and influential publications expounding advances in accuracy, speed, and form.

We then discuss important works in other related areas, particularly non-photorealistic rendering. Computer generated mosaics and collages have drawn upon the insights of several expositors of the art and science of NPR; we wish to catalog and summarize those works that have contributed algorithms, techniques, and points of philosophy to our own method.

2.2 Related Work

2.2.1 Art and Illustration

In this section, we examine the work of various artists whose interpretive and artistic processes have guided this work.

Collage Artists We have derived considerable inspiration from the work of Nick Bantock [4], a visual artist and painter who composes detailed collages as well as paintings and other types of art. His paintings and collages consist of objects and textural elements that are sometimes preserved whole, and sometimes used as combinations of fragments (figure 2.1). He produces both abstract and representational renderings.



Figure 2.1: “The Forgetting Room”, by Nick Bantock [4].

Similarly, the illustrations, graphic designs and artwork of Dave McKean [64] have been inspirational. In his unique style, he has used objects and textures together to create a strong atmosphere within each piece. Tending away from the strictly abstract, he combines well-known objects into unfamiliar and often shocking aggregations. He is able to manipulate the expectations of viewers, thwarting an initial interpretation by hiding additional meaning in the details. Figure 2.2 demonstrates McKean’s skill in richly layering textures, objects and symbols.

In the 16th century, Arcimboldo [21] produced surrealist portraits of human subjects represented as clusters of various types of objects, fruits being one notable example. In these portraits, both the human figures and fruits are easily recognizable (as in figure 2.3). Gal et al. [37] noted that this technique was given much more attention in the 20th century by artists such as Picasso, Miro, Dali, Chagall, Warhol, Rauschenberg, and others.



Figure 2.2: Detail from the graphic novel *Arkham Asylum: A Serious House on Serious Earth*, illustrated by Dave McKean [64].

Artists of the Collage, Mosaic and Double Image Forms In chapter 1 we discussed the double image form and examined works produced by prominent artists. Some other well known artists of double images are Sandro Del-Prete, Ken Knowlton, Vik Muniz, Ocatavio Ocampo, and Istavan Orosz; see Seckel’s book *Masters of Deception* [84] for a more complete treatment of this topic.

Chuck Close [89] is a well-known painter of photo-realistic portraits. Interestingly, his most recent portraits have been painted in a style that could be described as photo-realism when viewed at a distance but are actually composed of collections of discrete cells, each containing a small, simple, and abstract mini-painting. When viewed up close, the effect is entirely different and decidedly non-photorealistic. The cells are arranged in various different lattices and partition patterns, including rectangular, diamond shaped, and radially symmetric. Luong et al. [60] have reproduced Close’s style with a special filter, using isoluminant color pairs.

Salvador Dali [20], the surrealist painter, contributed an interesting work which may be described as a mosaic representation embedded within the environment of a more general surrealist painting. In this case, Dali wove another level of detail into the already multilayered mosaic form. This painting, titled “Gala contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln (Homage to Rothko)” features a likeness of Abraham Lincoln, rendered in blocks, behind the figure of a woman; the head of the woman blends into the eye of Lincoln. In fact, the block-rendering of Lincoln is an interpretation of that produced by Leon Harmon [8] , who completed several portraits in a heavily pixelated form. Battiato et al. [8] speculate that this is the first painting to use mosaic cells that each have a full range of tonality.



Figure 2.3: The portrait “Vertumnus” by Arcimboldo [21]

2.2.2 Image Mosaics

In this section we discuss the most important development in the science of computer-generated image mosaics, composed from libraries of images.

Early Work Exploration of computer generated image mosaics began with Silvers [86], in his picture book *Photomosaics*, published in 1997. As is noted by Battiato et al. [8], the techniques used to create the mosaics in the book has not been revealed, although it does not appear to employ technology that has not been duplicated or superseded in subsequent research efforts. The following year, Finkelstein and Range [36] produced an academic treatment of the image mosaic technique; figure 2.5 shows one of their mosaics. Previous to this, various artists had manually created photographic mosaics and photographic collages, producing dynamic works of art, but at a high time and labor cost. A survey of exploration of the photographic mosaic technique may be found in Battiato et al. [8].



Figure 2.4: “Gala contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln (Homage to Rothko)” by Salvador Dali [20]

Image Mosaics, Range and Finkelstein Finkelstein and Range [36] address the problem of creating a large scale image from a mosaic of smaller images that are arranged on a regular rectangular grid. The purpose is to find an arrangement of small images such that the effect of viewing the arrangement from a larger distance is that the images come together to form a larger image. They also explore methods of modifying the colors of the smaller images so that the mosaic better resembles the target image.

For this, they refer to the half-toning literature for inspiration. Also, some motivation for the image mosaic work comes from the visual principles exploited by impressionist painters; the human vision system has the tendency to combine multiple discrete features into single averaged features at a distance. They refer to manual productions of image mosaics by traditional artists such as Dalí and Close.

They have organized the mosaic creation into the following work-flow: choice of database images and constraint image; and choice of tiling and match finding for each tile (we have adopted a similar

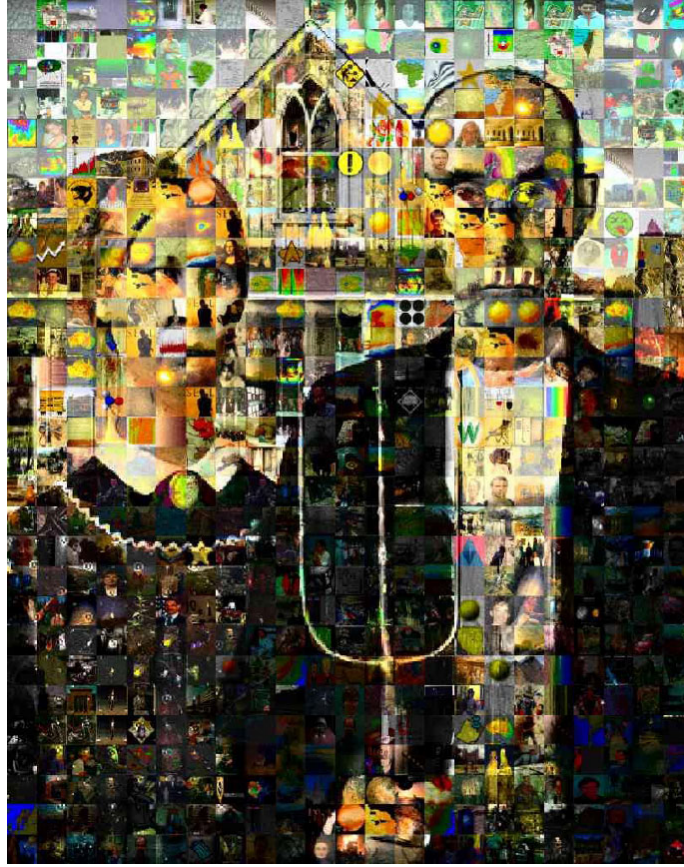


Figure 2.5: An example of Finkelstein and Range’s Image Mosaic [36].

work-flow for our own method). They declare the regular rectangular tiling as a natural choice, indicating that optimizing a mosaic over all possible tilings is too challenging of a problem. Indeed, neither do we attempt to optimize the mosaic over all tilings, but seek only to choose a tiling for which a very good mosaic is likely to exist (we define our conception of *very good* in chapter 4). They note that, based on the work of Ulichney [94], grids angled at 45% are the least distracting to viewers. However, it is not their intention to hide tile boundaries. We, however, *do* intend to hide tile boundaries, although we do not use a regular grid.

Finkelstein and Range consider several options for matching each tile to an image, including random matching, repeating the same image for each tile, manual human selection, mean tone matching, and matching upon more detailed image criteria. For our project, we have chosen the latter of these options.

In order to efficiently search the database of images for appropriate matches, they precompute a special set of wavelet coefficients, based on the method of Jacobs et al. [50]. This allows them to search a large database for a small number of terms; they report search times of below one second for a database of 20,000 images on a conventional computer.

Finkelstein and Range make some observations as to which images are most effective in composing photo mosaics. For the constraint image, they note that easily recognizable images, such as famous scenes, portraits or iconic images, as well as images that are recognizable at low resolutions, are desirable.

Inspired by the results in the halftoning literature, Finkelstein and Range seek to further improve the quality of the mosaic, after the matches have been obtained, by correcting the colors of each tile. They propose a correction function that adjusts the mean color of the tile to that of the constraint image by shifting and scaling the colors.



Figure 2.6: A cut-out image mosaic from the paper by Orchard and Kaplan [69]

Cut-Out Image Mosaics Orchard and Kaplan [69] have improved on the mosaic results obtained by Finkelstein and Range by introducing several process improvements. The most notable of these improvements are the use of image subregions, rather than entire images, global optimization of the match arrangement, and arbitrarily shaped tiles. These are considerable improvements over previous methods which sought only to compose mosaics of regular tiles containing whole images. Figure 2.6 shows one of their notable results.

Orchard and Kaplan describe the philosophy behind the image mosaic concept in the following excerpt [36]:

Image mosaics communicate at two disparate scales. These two scales act as a symbolic divide, so that the target image is conceptually set apart from the contents of the tiles that comprise it. This dichotomy provides a rich environment for combining images that either suggest the same message from two different perspectives, or supply contrasting viewpoints. For example, an image of a car might be made out of pictures

of the employees that manufactured it, or pictures of bicycles. Image mosaics are a powerful medium for conveying such split-level messages.

We attempt to bridge the divide between Orchard and Kaplan’s “two disparate scales” by producing image collages in which two or more scales are linked through a continuum. Specifically, we wish for semantic information to not be segregated by scale but to span the discrete scale levels of the mosaic.

Orchard and Kaplan go on to describe the tug-of-war between accuracy and discernibility when matching tiles with image regions. They resolve to seek the simultaneous fulfillment of accuracy and discernibility. That is, they wish to accurately represent the source image while also ensuring that the contents of each tile image are clearly discernible. The result of increased effectiveness in this goal, they state, is that fewer, larger tiles are needed to produce a high quality mosaic.

Orchard and Kaplan take a unique approach to accelerating the matching process. They point out the fact that the size of the database may be effectively extended by allowing sub-regions of each image to be used, rather than the entirety of each image. Alternately, they allow for rotations of each subregion, rather than requiring that the image be matched and composited in its original orientation.

They point out that it is impossible to achieve the best possible match while only matching coarse features as do Jacobs et al. [50] and Finkelstein and Range [36]. They propose a method for calculating a match score, based on SSD error (Sum of Squared Distance error), for each image in the database, at full resolution, for every possible spatial shift and rotation. In addition to all this, they calculate an optimal color correction as part of the match calculation.

Orchard and Kaplan implement this ambitious matching scheme by expanding the terms of the quadratic error metric, transforming the images into the frequency domain, and using spectral-based image registration techniques to quickly compute the error for every possible shift. In determining the final image for each tile, rather than using the match configuration with least error for each individual tile, a minimum cost result is globally computed for the entire tiling (according to well known graph-theoretic methods), so that no two images are repeated.

For future work, Orchard and Kaplan recommend exploring more transformation parameters for the image regions and feathered, non-binary tile masks. We have chosen to take up the exploration of the latter of these.

3D Collage Gal et al. [37] move the image mosaic into three dimensions; this transition requires them to explore not only superficial similarities between the color and lighting of images, but also more expressive correspondences, particularly in shape and volume. Correspondences between the shape of the constraint object and database objects are exploited, often to creative effect. The work has been inspired by the paintings of 16th century artist Arcimboldo.

As design objectives, Gal et al. attempt to match the shapes of objects while maintaining the following: a high proportion of intersection between the library and target object, with few protrusions; a low amount of intersection between adjacent library objects; and as little deviation as possible from the original size proportions of the objects in relation to each other. To compute match fidelity, they use a biased form of Euclidean distance on a voxelized 3D grid, as well as partial shape matching for localized regions. See figure 2.9 for a sample of their results.



Figure 2.7: A Jigsaw Image Mosaic [51].

Jigsaw Image Mosaic Related to the 3D collage, but back in the domain of two-dimensional image manipulation, is the Jigsaw Image Mosaic (or JIM), by Kim and Pellacini [51], which seeks to represent an image by a mosaic of irregularly shaped objects (figure 2.7). This work is less related to the image matching methods of the photographic mosaic and texture synthesis work, and more in line with the non-photorealistic, painterly, and tile mosaic rendering techniques that came before. Rather than drawing from an image database, the JIM method uses a database of shapes of relatively constant color, trimmed so that the boundaries of the shape match the boundaries of the represented object. The constraint image is segmented into regions of constant color, and then a packing is found to fill the regions with shapes from the database.

This whole process was optimized and improved by Di Blasi et al. [9]. Specifically, they increase matching speed by mapping the shape and color of each tile object into a metric space, so that they may be placed in an antipole data structure for easy searching. Shape data is represented by a vector consisting of the normalized distance of 90 object edge points from the object centroid.

Other Work Zhang [99] performs a comprehensive exploration of Content Based Image Retrieval methods for matching tiles to images from a database. She explores color and texture features, as well as histogram methods for creating searchable image signatures. She also employs histogram

methods for evaluating mosaic quality. Through these methods, she is able to significantly accelerate the production of accurate image mosaics. However, there are few or no cases in which both discernibility and accuracy are simultaneously handled; that is, it appears that if the image in each tile were replaced with a flat shaded region of appropriate color, the accuracy of the collage would hardly be affected.

Di Blasi et al. [10] propose a fast method for searching an image database by clustering image feature vectors according to an antipole tree structure. They use a simple three by three grid of mean values as the feature vector. They also introduce mosaics with multiple tile scales by partitioning the image according to a quad-tree; the effect is that lower detail regions of the image are represented by larger tiles, and the tile sizes vary by factors of two. Matching image details on a finer scale, however, is not treated; neither is the case of image-scale variation outside of even-factor increments.

Tran et al. [92] perform an evaluation of two photo mosaic algorithms. Along with computational efficiency, they measure the relationship between the following quality factors: similarity, granularity, variety, and database size. They evaluate two string matching algorithms; one that measures direct pixel-wise distance between the mosaic and original image, and another that uses a string alignment algorithm, treating each row of pixels as a string. They evaluate the quality of a mosaic by the minimum viewing distance at which no difference between the mosaic and original image is discernible. It was found that the minimum distance increases linearly with the area of the tiles. A weaker linear relationship was found to exist between the size of the image database and the similarity between the collage and the original image.

2.2.3 Collages

Digital Tapestry In their paper “Digital Tapestry”, Rother et al. [77] look at the problem of aggregate image composition from an entirely different angle. While the image mosaic works focus on finding an arrangement of images that well represents a target image, the tapestry problem is more focused on the quality of the arrangement and selection of database images, with the only large scale constraints being the salience of the image regions used, and the desirability of the aggregation to the eye of the user. They adapt a variety of methods, including methods similar to those used in texture transfer, to the task of choosing an aggregation that is representative of the database and arranged in a spatial layout that follows key design principles. See figure 2.9 for an example of this technique.

Autocollage In Autocollage, Rother et al. [78] present another, more sophisticated method for creating image collages which are not meant to approximate or represent any constraint image, but only to create an attractive arrangement of images that is representative of the image database



Figure 2.8: An example of a collage created by Autocollage [78]

(figure 2.8). They focus on creating collages in which each element is representative of the overall set. From each element image a region of interest is extracted. Certain classes of objects within the images are recognized, such as faces and skies, and the final packing of images is chosen so that certain relationships between these objects and the collage as a whole are respected. Finally, transition seams between images are rendered so as to not be obviously visible.

The collage problem is, in this case, proposed as a problem of finding a labeling that has minimum energy according to an energy function weighting library-representation, desirability of transition between images, proportion of region of interest used, and object-semantic considerations, such as faces and skies. The first two energy terms are minimized one their own, in order, and then the images are packed so that the entire collage is covered but regions of interest do not overlap. The packing is achieved via branch and bound and local searching techniques. Overlapping in the resulting packing is resolved through a graph cuts algorithm, as in Agarwala et al. [1] and Boykov et al. [11]. Alpha blending is achieved via alpha-Poisson blending, as in Pérez et al. [72], taking into account edge information so that a harder boundary is maintained along an edge and a softer transition is employed for smoother regions of the image. We employ a similar philosophy for blending between images.



(a)



(b)



(c)



(d)

Figure 2.9: The results of various collage and mosaic rendering algorithms. **2.9a** Digital Tapestry [77]. **2.9b** Smart Ideas for Photomosaic Rendering [10]. **2.9c** 3D Collage: Expressive Non-Realistic Modeling [37]. **2.9d** Puzzle Image Mosaic [9].

2.3 Literature Review

We will now briefly survey various other contributions that have influenced the direction of this project.

2.3.1 Mosaics

The photocollage art form shares some characteristics with the mosaic art form, which has been used by artists from classical times to the present [8]. We will summarize the history of the mosaic art form from the traditional forms to the computer-generated mosaic which has appeared more recently.

Battiato et al. [8] have compiled a survey of collage work until 2006, which the reader may refer to for a fuller history of traditional collage. They begin by discussing the works of Dali and Arcimboldo, two artists who combined pictorial elements to form larger scale images in the same vein as the modern photographic mosaic. They then discuss more contemporary artists, such as Harmon, Close, and McKean, who produced photographic mosaic-like images before software had been designed for the task.

Computer Generated Mosaics

Haelberli [42] conducted a wide exploration of painterly and NPR techniques for rendering images, including mosaic-like renderings that employ Voronoi tessellations and color the tiles with flat colors based on the source image. Dobashi et al. [25] expanded on this idea; they use Voronoi tiles to create a mosaic, but the tile centroids are iteratively moved so as to minimize the error between the mosaic and the source image. Later, Faustino and Figueiredo [35] also use Voronoi diagrams, but place the initial sites according to image features and shift the centroids iteratively towards a center of mass; they use a density function which is itself derived from image features—specifically, the gradient magnitude of the image.

Hausner [43] was the first to attempt an NPR simulation of classic tile mosaics, with considerable success as the reader can see in figure 2.10. In this paper, a process is described in which tiles are oriented along image features and are arranged so as not to straddle important edges. As in Faustino and Figueiredo [35], centroidal Voronoi diagrams are used with a modified metric to achieve the desired lattice and orientation information from the edges of the image is used to orient the tiles. Elber and Wolberg [32] extended this idea with a more sophisticated approach to tile orientation, relying on user defined parametric curves. Tiles are placed tangential to the parametric curves in concentric rows so that each tile is touching the one placed before it. Battiato et al. [7] improve Hausner’s technique by segmenting the image, via statistical region merging and user input, into foreground and background regions, similarly orienting tiles along image features and also providing

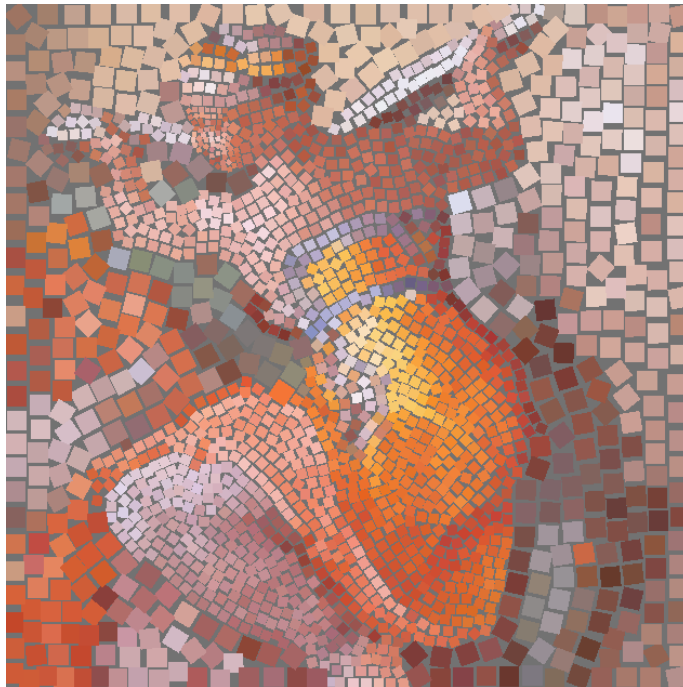


Figure 2.10: A mosaic by Hausner [43] exhibiting tiles of size and orientation adapted from the target image.

some contingency for tile overlap. The tiles are arranged according to the level set curves obtained from the distance transform, with the region edge map as input.

These techniques were further adapted and refined in Battiato et al. [6], who use gradient vector fields to determine tile placement and orientation and Liu et al. [58], who use a graph-cuts based global optimization scheme to obtain a globally optimal solution. Schlechtweg et al. [82] employs the novel approach of using renderbots, a multi-agent system, to simulate various NPR effects including mosaics.

2.3.2 Image Abstraction and Non-photorealistic Rendering

The problem of image abstraction involves the analysis and manipulation of images, particularly photographs, with a mind to suppress image details that are not immediately necessary for the perception of large scale object and semantic content [22]. The abstraction of images involves the reduction of the image data to a minimum set necessary for effective perception and understanding of the image content.

An area that is closely related to image abstraction is non-photorealistic rendering. Indeed, the process of NPR may, in many cases, be characterized as the separation of rendering into an abstraction stage, in which non-salient information is identified and discarded, and a stylization stage, in which new information is added to the image, allowing it to be recognizable as a work

belonging to some artistic tradition. Conversely, abstraction may also be thought of as a type of non-photorealistic rendering, since it may be used to convert a photograph or some other realistic image, into something more stylized. The motivation for techniques developed in this area have come from early work in psychovisual perception, such as Marr [62], and more recent work such as Regan [76].

We have approached the problem of the photocollage from the point of view of both image abstraction and NPR, in that we must make decisions about which information from the target image we wish to preserve and which we would like to replace with information from database images. We wish to produce a collage in which both the target image and each of the elementary library images are obviously recognizable to a viewer; but how do we know which portions from each image to use? We wish to ensure a match, wherever is necessary to achieve an appropriate level of accuracy, between low, mid and high frequency spatial frequency components. The image abstraction segment of this workflow comes in as we determine which regions of the image require matching at a high level of detail, and which may be approximated by mere coarse matches, and matches of larger scale structure.

Herman and Duke [45] provide a concise and eloquent description of the philosophy behind non-photorealistic rendering, summarizing the trajectory of computer graphics research towards realism up until that point and drawing on contrasting philosophies in traditional eastern and western art. They point out that impressionism is also an important artistic direction which deserves attention and provides considerable utility in many applications. The key difference between the approaches of the impressionist artist and the realist artist is that the former convey the impression of objects or a scene without presenting all of the detail that would be encountered when viewing the scene in reality; lines, contours, shading and shape are all used to build the general idea of an image. The image mosaic and the image collage may be viewed, in a sense, as impressionist forms, since in these cases we wish to give an impression of a target image through the shapes, textures and colors of an aggregation of other images.

That same year, Gooch and Gooch provided an important treatment of non-photorealistic rendering [39], which the reader may refer to for a summary of the field up until that time.

DeCarlo and Santella [22] provide a major examination of pure image abstraction, using a priori knowledge of image semantics obtained from eye tracking data. A scale-organized segmentation hierarchy is used to abstract non-salient areas of the image into flat shaded regions. Figure 2.11 shows one of their results.

Inspired by recent video abstraction endeavors in cinema, Winnemöller et al. [97] expanded the work of DeCarlo and Santella [22] into the domain of moving images, dealing with the issue of temporal coherence while streamlining the salience identification process and moving it offline and independent of significant user input.



Figure 2.11: **2.11a** The original image. **2.11b** The abstracted image via the method of DeCarlo and Santella [22] .

Orzan et al. [70] significantly improved the automation of image salience extraction for image abstraction by recognizing that image edges may be organized according to their position in Gaussian scale space and used as the main indicators of semantic information within an image. They adapt methods of gradient space image integration to recompose images from gradient edge data. As is evident in figure 2.12, they have succeeded in reducing the prevalence non-salient image structures.

Painterly Rendering

Viewed as an artistic rendering problem, several similarities emerge between the photocollage and painterly rendering techniques. The photocollage may be viewed as a type of painterly rendering filter; that is, a source target image is rendered in a particular artistic style, so that it resembles a piece that was composed by some creative method other than photography.

Haelberli [42] provided an extensive exploration of painterly rendering and other NPR techniques. Introducing brush strokes as the primary rendering primitive, each brush stroke being defined by a list of parameters such as stroke size and orientation, and locally controlling these parameters, he set the standard for painterly rendering and NPR techniques.

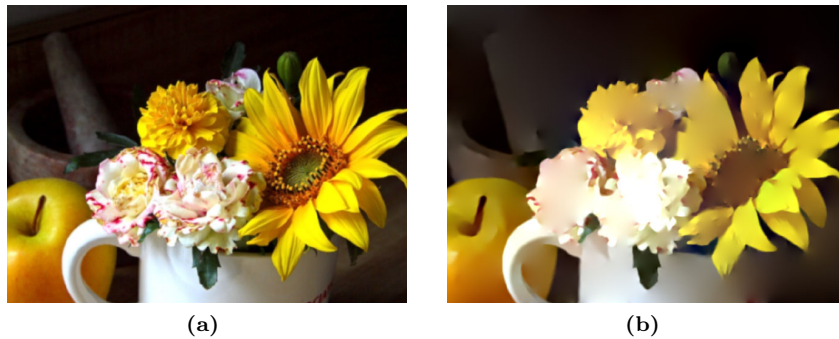


Figure 4.2: **4.2a** The original image. **4.2b** The abstracted image via the method of Orzan et al. [70] .

This work was followed up extensively by other authors, such as Litwinowicz [57] and Shiraishiv and Yamaguchi [85], which offer alternative methods for stroke orientation and placement, and Markosian et al. [61] who explore the acceleration of NPR rendering for real time applications and line rendering of geometric objects. Hertzmann [46] provides a method of building a painting as a series coarse to fine approximations with each successive layer replacing high-error regions of the previous layer. This work, in particular, has inspired the iterative refinement methods we use as part of our method. This iterative refinement process is illustrated in figure 2.13.



Figure 2.13: **2.13a** The original image. **2.13b** A painterly rendering via Hertzmann’s method [46] with a coarse brush. **2.13c** The previous rendering after refinement with a finer brush.

Hertzmann [47] provides a survey of stroke-based rendering, including painterly rendering techniques which the reader may refer to for a more detailed discussion. He unifies the many disparate theories of stroke-based rendering, expounded by various computer vision and graphics researchers of unrelated problems, in an attempt to compose a more generalized theory by which the greater philosophy of stroke-based rendering may be understood and advanced. He identifies the placement of render elements according to some variety of goals as a coarse formulation of stroke-based rendering; certainly, the problem of image mosaics and image collages may be viewed thus, since in these cases we seek to place images, which are the rendering primitives, in order to approximate some target image.

2.3.3 Texture Synthesis

The domain of texture synthesis and texture transfer has provided considerable inspiration for this work. The work done in the area of texture synthesis-from-example has focused on transferring the texture characteristics from a source texture image to the contents of a target image. The most successful results rely on finding matches between the target image and a texture exemplar.

Efros and Leung [30] were the first to employ neighborhood sampling for the composition of texturally coherent regions. At the same time, Wei [95] introduced a method for synthesizing textures from exemplars by analyzing the image and creating a deterministic texture function. These methods were expanded and refined by Wei and Levoy [96], who built on the idea that textures could be successfully replicated, without repetition, by copying pixels from an exemplar

texture; the constraint is that the neighborhood of each copied pixel should correspond in some way to a similar neighborhood in the example texture. They introduced performance optimizations, such as tree-structured vector quantization of the search space and pyramidal sampling of the example texture. These ideas were built upon by several researchers, including Ashikhmin [2] who improved user control, and modified the pixel selection methods of WL synthesis to overcome the shortcomings of oversmoothing. Ashikhmin recognized that the L2 distance between pixels values used by Wei and Levoy is an insufficient approximation of how the human visual system measures the perceptual distance between image elements; the human vision system is sensitive to corners and edges, and the L2 norm does not reflect this in any way. Ashikhmin solved this problem by relying not on the L2 norm to match pixels, but on the visual continuity of the exemplar image itself, using a verbatim copying strategy to reproduce texturally coherent image patches, the seams of which are naturally hidden within the activity of the texture.

Later, Efros and Freeman [29] modified this pixel-wise sampling procedure by instead copying patches of texture, rather than single pixels, in order to preserve intratexture coherence. Similar to Ashikhmin, they rely on the verbatim copying of texture patches to ensure that the character of the texture is not lost through the excessive fine-grained fragmentation of pixel-wise synthesis techniques. This presents the problem of visually disruptive patch boundaries. They devise a seam repair process in which a least cost boundary is computed between adjacent overlapping tiles. For the purposes of texture transfer, they introduce another constraint to the patch selection; in addition to ensuring that the neighborhood of the patch corresponds to the example texture, they require also that the low spatial frequency component of the patch must correspond to the low spatial frequency component of some constraint image. The result is an effective method of texture transfer, in which a texture may be replicated for arbitrarily large areas but also arranged so as to resemble some non-texture target image. This method for texture transfer, illustrated in figure 2.14, suggests a feasible model for photocollage composition, and we have drawn inspiration from it. Specifically, we have adapted the methods of match acceleration, seam repair and match calculation which are the basis of this method. The photocollage, however, differs significantly from texture transfer, and special considerations of these differences has been required in adapting these methods. For instance, texture image data below a certain cut-off spatial frequency is often irrelevant, in that it contributes very little to the perceptual characteristics of the texture; this is not the case for the library images from which we compose the photocollage. The process of texture transfer is, perhaps, more akin to that of the image mosaic, since coherent image patches are copied in order to match a target image, although the image patches come from many different images in the case of the image mosaic rather than a single texture.

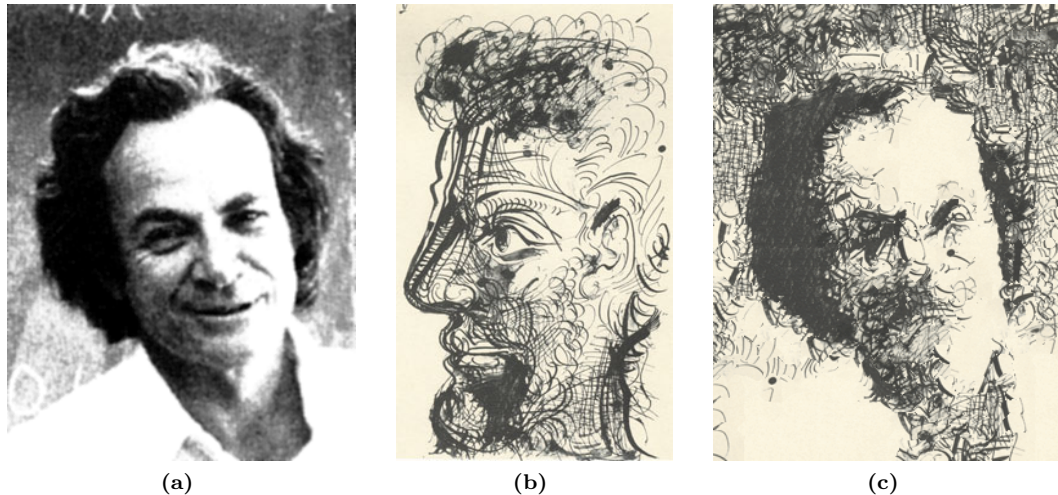


Figure 2.14: **2.14a** The target image. **2.14b** The exemplar texture, to be transferred to the target image. **2.14c** The texture transfer result, via the method of Efros and Freeman [29].

2.3.4 Image Completion

Image completion algorithms seek to replace a region of an image, often determined by the user, with different content than that which occupied that region originally; this is done either based on other content in the same image, or content from some other image or library of images. The photocollage problem may be thought of as a series of constrained image completion steps on a partition of the image.

Criminisi [19] and Drori [26] use patch copying to fill in user-segmented regions according to the content of the rest of the image. They combine techniques from texture synthesis and structure-based inpainting. Diakopoulos [23] uses the texture synthesis ideas previously discussed to resynthesize user selected regions according to semantically coherent content chosen from a user-annotated database. Komodakis [55] uses global optimization, specifically Priority Belief Propagation, to improve the greedy approach taken by the previous methods.

Hays and Efros [44] proffer a solution to the problem of searching a very large unannotated database for a desirable patch; they adapt methods of semantic content based image clustering, originally devised by Oliva and Torralba [67]. In this way they are able to eliminate most choices from a very large database. They further refine template matching techniques for the problem of scene completion. Oliva and Torralba use coarse global features to automatically classify images with semantic tags. Both teams find that the process of semantic clustering is effective over very large image databases. For image completion, this means that the improved accuracy afforded by a very large database needn't come at such a high search cost. The results of Hays and Efros's work are illustrated in figure 2.15

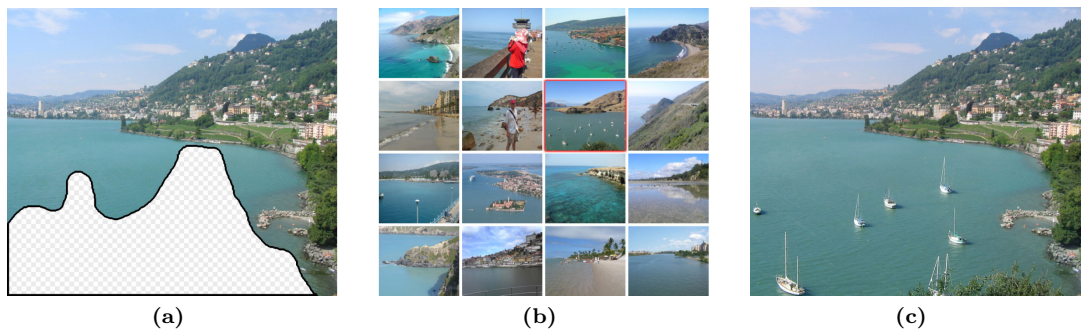


Figure 2.15: A demonstration of Hays and Efros's method [44] for scene completion. **2.15a** The target image, with the replacement region flagged. **2.15b** A small subset of possible replacement images. **2.15c** The final composition.

2.3.5 Frequency Space Image Synthesis

Oliva and Torralba, in their paper “Hybrid Images” [68], create images that are composed of data that are semantically coherent only within certain bands of spatial frequency. Specifically, they compose the low frequency data and high frequency data of two separate images in order to create a hybrid image that presents different content depending on the viewing distance (figure 2.16). This idea is similar to the idea of image collages, except that image collages, upon close viewing, represent an aggregation of images rather than a single image. Oliva and Torralba characterize this difference as such: image mosaics contain a local and global interpretation, whereas hybrid images contain two distinct global interpretations.



(a)

Figure 2.16: A hybrid image, by the method of Oliva and Torralba [68]. After examining the image up close, take a few steps back and look again.

2.4 Summary

In this chapter, we have discussed all of the important works which have inspired, informed and guided this work. We began by examining the artwork of Nick Bantock, Dave McKean, and others, whose styles we have attempted to imitate and reproduce. We then discussed important milestones

in the progress of image mosaic methods and collage methods. Each of these composition systems represent a simpler or slightly different version of the problem we address in this thesis, and thus we seek to combine as many of these methods as we can. Particularly, we have borrowed from these previously described systems of signature extraction and matching, color correction and color shifting, spatial shifting, region-of-interest extraction, and tile boundary hiding.

Finally, we surveyed a selection of works in the areas of texture synthesis and non-photorealistic rendering in order to gain insight from previously described solutions to related problems. Particularly, we have been interested in techniques for image abstraction and content summary, stroke based rendering, patch-based rendering, texture analysis, and multiscale rendering and image composition.

CHAPTER 3

TEXTURE AMPLITUDE AND SALIENCE MEASURES

3.1 Overview

In chapter 1 we set out a general outline for the discussion of our photocollage algorithm. Recall that the first step is the preprocessing of the images in the library, done first to allow fast image loading for all subsequent collages. In this step we downsample each image and extract texture and edge information for subsequent image analysis operations. In this chapter we describe in detail the techniques we use to obtain these texture and edge maps from each library image.

3.2 Requirements for a Texture Contrast Map

In chapter 4, we will include texture contrast as an image feature to be considered as part of an image comparison and matching strategy. In this case, we need a fast method for calculating a texture contrast map for each image in the library. Also, in chapter 6 we will use a texture contrast map as part of a proposed image abstraction scheme, using also a bilateral filter with depth controlled by the contrast of the textures. For a texture contrast map to be useful in these tasks, it must have certain properties.

We require a texture contrast map that shows low values for texture regions with low local contrast and high values for textures with high local contrast. The meaning of “local” in this context refers to the scale at which the texture contrast map is computed. We say a texture has high contrast, for a given scale, when there is a large difference between the values of the local extrema within a region of that scale. However, the operation should be scale invariant in some sense, and should respond to textures of varying scales, within some range of scales. That is, a single texture contrast map, for some chosen scale, should sufficiently account for the majority of significant texture activity in an image. The map should not be affected by non-texture features such as edges. The map should show roughly uniform texture contrast values for regions of uniform texture; it should map unsmooth regions to smooth regions, according to the character of the unsmooth regions, or regions of homogeneous texture to regions of homogeneous luminance.

Bae et al. [3] introduced a quantity called *textureness* that is similar to texture contrast. For an

image I , they measure textureness by taking a highpass filter $H(I)$ and then filtering its magnitude $|H|$ bilaterally with I . They compare this result to the *activity/power map* of I , which is the Gaussian lowpass filter of $|H|$; the activity/power map effectively measures texture activity in an image, but is spatially imprecise and suffers from haloing (“haloing” refers to, in this case, an artificial glow radiating from high activity regions). The textureness map, on the other hand, is more spatially precise because the bilateral filter is constrained by the features of the original image; so, if a texture region is bounded by a step edge, the bilateral filter will ensure that the high response of the textureness map will not flow over the step edge. However, this textureness map does not satisfy our condition that regions of uniform texture be represented in the texture contrast map by regions of uniform texture contrast (see figure 3.9). The spatial precision comes with a loss of homogeneity within texture regions. However, one may construct a similar filter by replacing the Gaussian highpass result $|H| = |I - G(I)|$ with a filter $|H_m| = |I - M(I)|$ (similar to a highpass filter, but subtracting a median filter operation rather than a Gaussian filter) and complete the operation with a median filter rather than a bilateral filter (figure 3.1). The advantage of this replacement is that the resulting texture contrast map is smoother, and still respects edges. The disadvantage is that some spatial precision is lost due to the propensity of the median filter to obliterate corners and other fine-scale features. We have, however, found a way to overcome this limitation which we will discuss shortly.

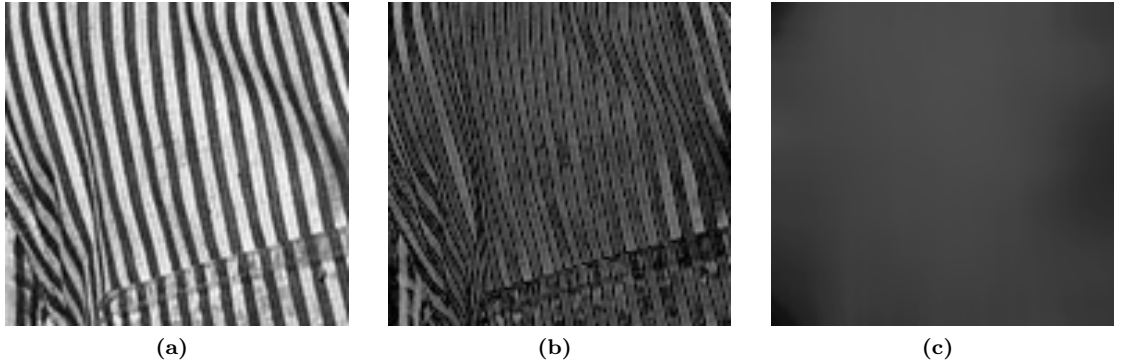


Figure 3.1: **3.1a** Texture image I . **3.1b** $|H_m| = |I - M(I)|$ where M represents the median filter for some radius. **3.1c** $M(|H_m|)$.

3.3 Properties of Texture Images

Of course, in order to go any further with an operation that is able to adequately process texture images, we need to decide what we mean by “texture”. The concept of texture is difficult to define, but yet most people have some intuitive sense of what it is [93]. In order to add precision to the idea of texture, we refer to the definition provided by Sklansky [87], which identifies constant, periodic,

or slowly varying statistical parameters as an identifying characteristic of textures. From this, we can posit the existence of a sort of statistically perfect texture that has uniformly identical statistical parameters everywhere inside the texture region. For simplicity, we may say that the statistical parameters of interest are any parameters that may be derived from a gray level histogram; taking it one step further we may say that, for every neighborhood of some scale within the texture, all histograms must be identical. Then, what we have done is specified a class of images that are like textures, or have texture-like properties. This has been useful in formulating a general operation for measuring texture contrast.

3.4 A Basic Texture Contrast Map

The motivation to isolate this particular class of texture-like images has come from the desire to improve the method of Bae et al. Naturally, if every subregion of scale r of a texture I has an identical histogram, then the median filter $|M(I, r)|$ over the texture will be uniform and constant, since the median values may be extracted from the histograms. We may then perform the operation $H_m = I - M(I, r)$ in order to extract the texture details from the rest of the image content. Keep in mind that the regions of texture in I are also regions of texture in H_m , since $M(I, r)$ is uniform across texture regions, and thus all histograms in H_m are still identical. Also, these new textures in H_m will have similar contrast to the textures in I , since all we are doing is recentering these textures about 0. So, we may then perform a final median filter on $|H_m|$, to obtain $M(|H_m|, r)$, a smooth texture activity map. Note that, again, regions of texture in I will be mapped to regions of uniform value in $M(|H_m|, r)$. Assuming that the median filter value of $|H_m|$ is proportional to the contrast of H_m , we then have an effective texture contrast operation τ_r :

$$\tau_r(I) = M(|I - M(I, r)|, r), \quad (3.1)$$

where $M(., r)$ denotes the median filter for some radius r . (we discuss cases in which this assumption does *not* hold later in the chapter). Figure 3.2 shows the stages in the production of this texture contrast map.

3.5 A Better Texture Contrast Map

Returning to the idea of the “statistically perfect texture”, we must acknowledge that the real textures we encounter in images do not conform to this strict model. In practice, textures are complicated and may not have any determinable set of statistical parameters that are identical everywhere for any particular scale. Therefore, our texture contrast map will be similarly imperfect, though still useful. Also, while these criteria for texture-like images seem to be suitably general to

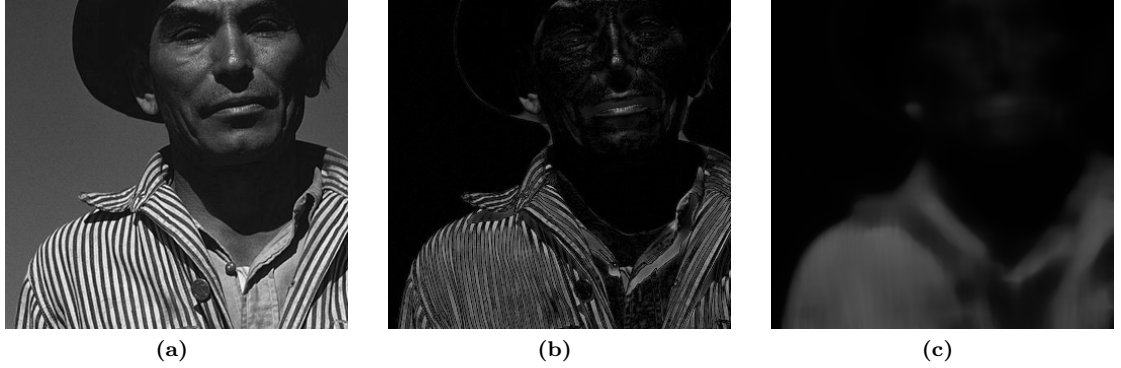


Figure 3.2: **3.2a** Image I . **3.2b** $|H_m| = |I - M(I, r)|$. **3.2c** $M(|H_m|, r)$.

encompass a wide range of textures, there are some textures that are not accounted for. This is compounded by the fact that our assumption that the median value of a region is proportional to the contrast of the region does not hold for every texture. These assumptions do hold for textures that adhere to a distribution that is, in some sense, “well behaved”; that is, for distributions that are more or less unimodal (although this is not a strict requirement). These assumptions fail for textures that have eccentric distributions of luminance values. While there exist textures that will not hold to this assumption, we have found that it is sufficient for a large class of naturally occurring textures, as we will show in section 3.5.2 (see figure 3.6).

As for our texture contrast operator τ_r , we have seen that while it provides a good measure of texture contrast, it is spatially imprecise, since its precision is limited by the radius r of the median filter. Of course, one may increase the spatial accuracy of the filter by decreasing the radius r , at the cost of measuring a set of textures of a higher frequency. However, we may combine texture contrast maps of various radii in order to derive the benefits of low-radius, spatially precise filters, while maintaining the robustness of high-radius filters. At this point, we bring attention to the effect of iteratively applying the operator τ_r .

3.5.1 Histogram Folding

Consider what is happening to the histogram of an image when applying this filter. By taking the absolute value of the median difference filter $|H_m| = |I - M(I)|$ we are “folding” the histogram of I about the value $M(I)$; values of I that were previously equal to $M(I)$ become 0, and all values which previously lay below $M(I)$ are added back into the histogram in reverse bin-order. So, the histogram is shifted left by $M(I)$ bins, and all bins with value less than $M(I)$ are added back into the histogram in reverse order, left to right. This process is illustrated in figure 3.3.

Precisely, for a histogram Hist of image I with median value $M(I) \in [0, 1]$, $M(I) > 1 - M(I)$, the histogram Hist_m of image $|H_m|$ is as follows:

$$\text{Hist}_m(b) = \begin{cases} \text{Hist}(M(I) + b) + \text{Hist}(M(I) - b) & \text{for } b \in [0, 1 - M(I)] \\ \text{Hist}(M(I) + b) & \text{for } b \in [1 - M(I), M(I)] \end{cases} \quad (3.2)$$

Similarly, in the case that $1 - M(I) > M(I)$, we have:

$$\text{Hist}_m(b) = \begin{cases} \text{Hist}(M(I) + b) + \text{Hist}(M(I) - b) & \text{for } b \in [0, M(I)] \\ \text{Hist}(M(I) + b) & \text{for } b \in [M(I), 1 - M(I)] \end{cases} \quad (3.3)$$

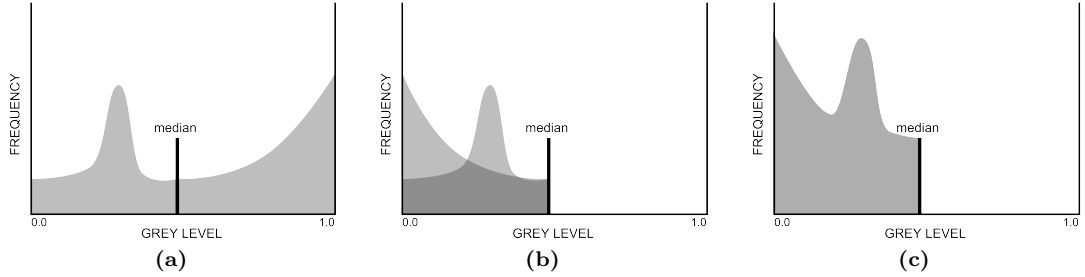


Figure 3.3: One iteration of the histogram folding process, which occurs during one iteration of operation k . **3.3a** Original histogram. **3.3b** Histogram with median indicated, and top end folded over the bottom end. **3.3c** Folded histogram.

The upshot of this histogram folding process is that the maximum value of $|H_m|$ is equal to either $1 - M(I)$ or $M(I)$, both of which are less than or equal to the maximum value of I . As the lowest bin fills with each iteration, the median value will eventually be zero; at this point, folding will no longer occur. For simplicity we define the n th iteration of this operation as follows:

$$I_n = |I_{n-1} - M(I_{n-1})|. \quad (3.4)$$

It is important to note that wherever $M(I_n) = 0$, histogram folding will not occur, because nothing is being subtracted from the image for the next iteration. This is important because while $M(I_n)$, for some sufficiently high number of iterations n , will go to zero, I_n itself will not necessarily go to zero. We discuss the importance of this observation later when discussing the *corner map*.

At this point, the important observation is that with each additional iteration $\tau_n = M(|I_{n-1} - M(I_{n-1})|) = M(I_n)$, the magnitude of τ_n is decreased from the magnitude of τ_{n-1} , as illustrated in figure 3.4. We may exploit this to compose a new, spatially precise texture contrast filter that consists of the sum of texture contrast filters of decreasing radius and decreasing magnitude. With each additional term in the sum, corners are filled in more tightly and the final map is more representative of the content of the image I . So, our final texture contrast operator may be expressed as follows:

$$\tau_{\text{final}}(I, r) = \sum_{n=0}^N \tau_n(I, r/(2^n)), \quad (3.5)$$

where $\tau_n(I, R)$ represents the operation $\tau(I, R) = M(|I - M(I, R)|, R)$ iterated n times. Figure 3.4 shows the individual terms in this sum, and the final result.



Figure 3.4: **3.4a–3.4c** A view of each term in the sum of images used to compose the final, spatially precise texture contrast map. Note that while each increasing term is reduced in amplitude, its spatial precision is increased. The first coarsest scale is radius=6, and decreases by half at each iteration. **3.4d** Original image. **3.4e** Final texture contrast map.

3.5.2 Evaluation

We have tested this texture contrast map on 112 of the standard Brodatz textures [12]. By inspection, we have classified the results of the texture contrast operation on each texture according to its success in properly representing contrast. The operation was said to be successful for a given texture if the resulting map values were relatively uniform, according to a qualitative perceptual judgment, and approximately proportional to the perceived contrast of the texture. The operation was said to be unsuccessful if the map values did not correspond accurately to the contrast of the texture. The operation was said to be *partially* successful if the map values were somewhat faithful to the perceived contrast of the texture, but were not uniform or showed clustering. Also, some of the Brodatz textures do not meet our definition of texture because they do not exhibit the required statistical stationarity; we have discounted results from these images. All judgments were made according to the perception of the experimenter.

We observed that 67 of the 112 Brodatz textures, or 60%, are successfully measured by our algorithm; 27 textures, or 24% are measured with partial success; 13 textures, or 12%, are *not*

successfully measured; and 5 textures, or 4%, do not meet our requirements. See figure 3.6 for a selection of textures that are successfully and unsuccessfully measured. Figure 3.5 shows a selection of images and their texture contrast maps.



Figure 3.5: Two images (3.5a and 3.5c and their texture contrast maps (3.5b and 3.5d).

Comparison with Other Methods

Texture classification has been widely studied for several decades, and thus many methods exist for deriving various features from a texture image. Here, we will compare our texture contrast measurement with a few other well known measurement methods.

Carson et al. [14] have developed a system for querying a database of images based on image segmentations that seek to partition each image according to regions of roughly constant color and texture. To this end, they develop a measurement of texture contrast that uses information derived from a windowed second moment matrix of the gradient magnitude edge map, and a unique form of scale detection. This scale detection is performed by measuring the difference between scales of the polarity of the image at each pixel, where polarity refers to the tendency of gradient vectors in the neighborhood to point in the same direction. The texture contrast itself is derived as a quantity

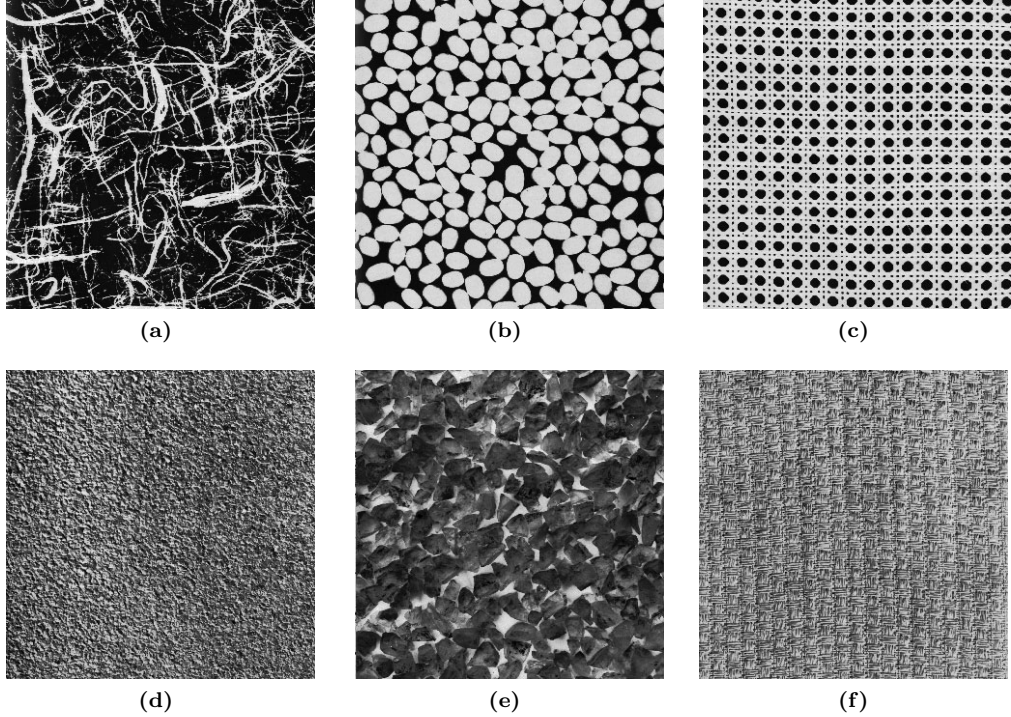


Figure 3.6: A sample of Brodatz textures [12] that do and do not agree with our assumptions. The top row of images, figures **3.6a-3.6c**, which would generally be classified as textures by a human observer, unfortunately do not have properties which allow them to be classified as textures by our method.

composed from the eigenvalues of the second moment matrix. For our purposes, it is not necessary to monitor the scale of the textures we are measuring, since we are primarily concerned with the scale of the collage tiles, and thus wish to scale our measurements accordingly. However, we have found that when the appropriate scale is hand-chosen, and applied uniformly over the entire image, our method achieves similar or better fidelity to edges and other object details, where the method of Carson et al. produces some blurring and haloing, as we see in figure 3.9.

Lozano and Escolano [59] propose an alternative measure of texture contrast, informed by the method of Carson et al. but using entropy-based image measurements. Comparison of our method with theirs reveals the weakness of our method when applied to an image with textures of highly varying scales. While the method of Lozano and Escolano correctly classifies textures across a range of scales, we see in figure 3.7 that our method has ignored those textures larger than the specified radius. When the radius is increased, textures of all scales are accounted for, but at the expense of object edge fidelity. However, our method does provide a smoother representation for a given scale, and is not hampered by the artifacts observed around edges in Lozano and Escolano’s effort.

Kokkinos et al. [54] use Amplitude-Modulation-Frequency-Modulation (AMFM) methods to extract a texture feature vector, and Dominant Component Analysis (DCA) to distill a set of basic

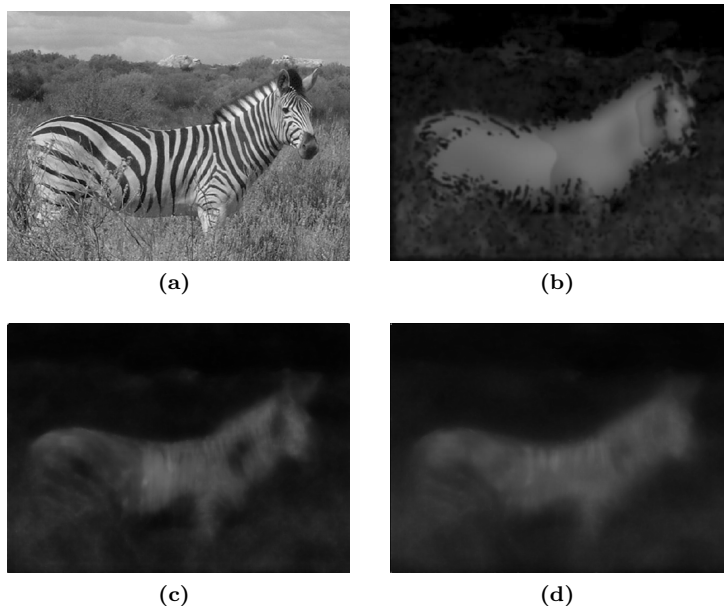


Figure 3.7: **3.7a** Original image. **3.7b** Texture contrast by Lozano and Escolano [59]. **3.7c** Our texture contrast map calculated with radius $r = 32$. **3.7d** Our texture contrast map calculated with radius $r = 64$.

texture descriptors, including texture contrast. As with the previous methods, the response of their filter to varying scales is superior. However, from examining figure 3.8 it is clear that the results, while useful for certain segmentation tasks, may not be ideal for smoothing filters due to map discontinuities.

The method of Bae et al. [3] for producing a “textureness” map may also be held up for comparison (figure 3.9), since it measures a similar quantity to texture contrast. In some ways, their textureness map is superior to all the previous methods, in that it is conceptually simple (it is built on a cross-bilateral filter) and produces results with very high spatial accuracy. However, it is corrupted by edges and other high-salience image features, and highlights texture features proportionally to contrast, rather than uniformly shading a region of coherent texture contrast.

3.6 Saliency Maps

One more loose end we must tie up is the interpretation of the map $I_n = |I - M(I_{n-1})|$ in the limit as $n \rightarrow \infty$ and where $I_0 = I$. As we have observed, for large enough n , $M((I_{n-1}(x, y))) = 0$ for all pixels (x, y) , even though $I_{n-1}(x, y)$ itself is not necessarily uniformly zero for all pixels (x, y) . We have discovered that it is *not* equal to zero for pixels that correspond to edges and corners in the image I , but equal to zero everywhere else as we see in figure 3.10. This is not surprising, since we have already seen that the histogram folding process causes the median map of textured regions to go zero with increasing iterations, indicating that textured regions are being reduced in magnitude

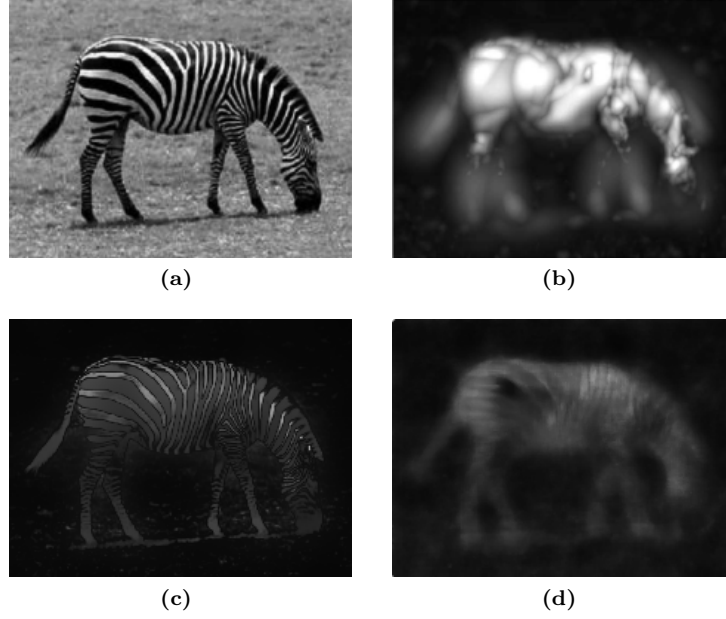


Figure 3.8: **3.8a** Original image. **3.8b** Texture contrast by Kokkinos et al. [54]. **3.8c** Textureness by Bae et al. [3]. **3.8d** Our Texture contrast map.

with each iteration. Then, any part of the image that remains greater than zero either is not a texture or has never contained any activity to begin with.

Because the map $\lim_{n \rightarrow \infty} I_n$ tends to highlight corners and other details that are suppressed by the median filter, we have called it the *corner map*:

$$\text{Corner}(I, r) = \lim_{n \rightarrow \infty} I_{n,r}, \quad (3.6)$$

where r indicates the scale of the median filter $M(., r)$.

Figure 3.10 shows the intermediate stages in the computation of the corner map as n increases.

3.7 Edge Maps

For our matching calculation, as we will describe in chapter 4, we require coarse-to-fine scale edge maps. We also use these edge maps as the mass density functions for the generation of centroidal Voronoi diagrams. In order to obtain this edge image F from a source image I , we first take a simple normalized gradient magnitude map of the Gaussian filtered image, with filter radius r appropriate for the desired scale. We will call this initial estimate ξ :

$$\xi_r = |\nabla G_r * I|, \quad (3.7)$$

where G_r represents a normalized Gaussian smoothing operation of radius r .

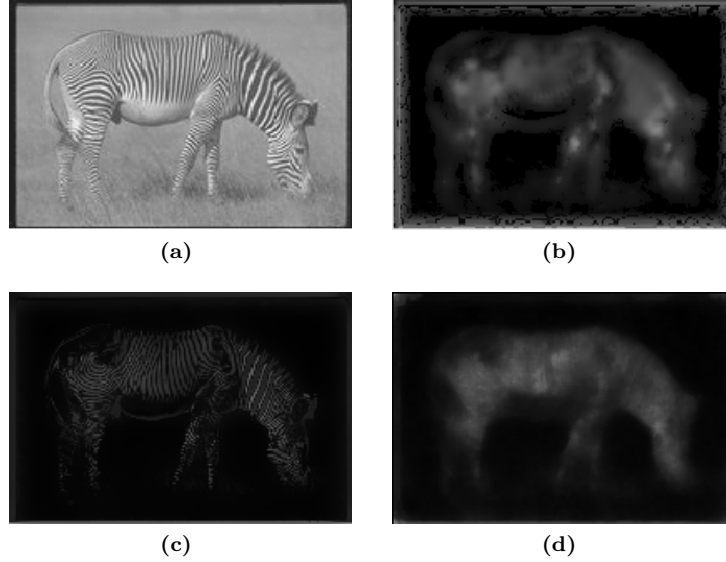


Figure 3.9: **3.9a** Original image. **3.9b** Texture contrast by Carson et al. [14]. **3.9c** Textureness by Bae et al. [3]. **3.9d** Our Texture contrast map.

This provides us with a coarse idea of where and how strong the edges are at the desired scale. We subtract the result from 1 so that edges are represented by lower values. This edge map will be contaminated by activity resulting from texture edges or non edge perturbations in the original image. We may remove this unwanted activity by taking the corner map, $\text{Corner}(\xi, r)$, of the edge map F .

$$F''_r(I) = 1 - \text{Corner}_r(\xi_r(I)), \quad (3.8)$$

The resulting map F'' appropriately represents edges and is uncontaminated by texture data (figure 3.11). This is desirable for both of the tasks which this map will be used in; for the match calculation, we wish to treat edges specially, but not textured regions; for the generation and placement of Voronoi centroids, we do not wish to have increased density or clustering of points within textured regions.

While it is possible to achieve similar properties in an edge map by taking the gradient magnitude of the median filtered image, our method is preferable because it decouples the edge scale from the texture scale; the former method requires more aggressive median filtering to remove larger scale textures, and thus the accuracy is reduced for finer scaled edge maps.

Looking forward to our intended use for this edge map in the matching process, it is important to recognize that the edge map for a particular scale should contain data from each scale below it. We will use this map to prevent over-refinement of the collage near image edges. So, if the edge map didn't contain data from a wide range of scales, over-refinement would be allowed to occur during the refinement passes (see chapter 4 for a more detailed discussion of this process). If errors

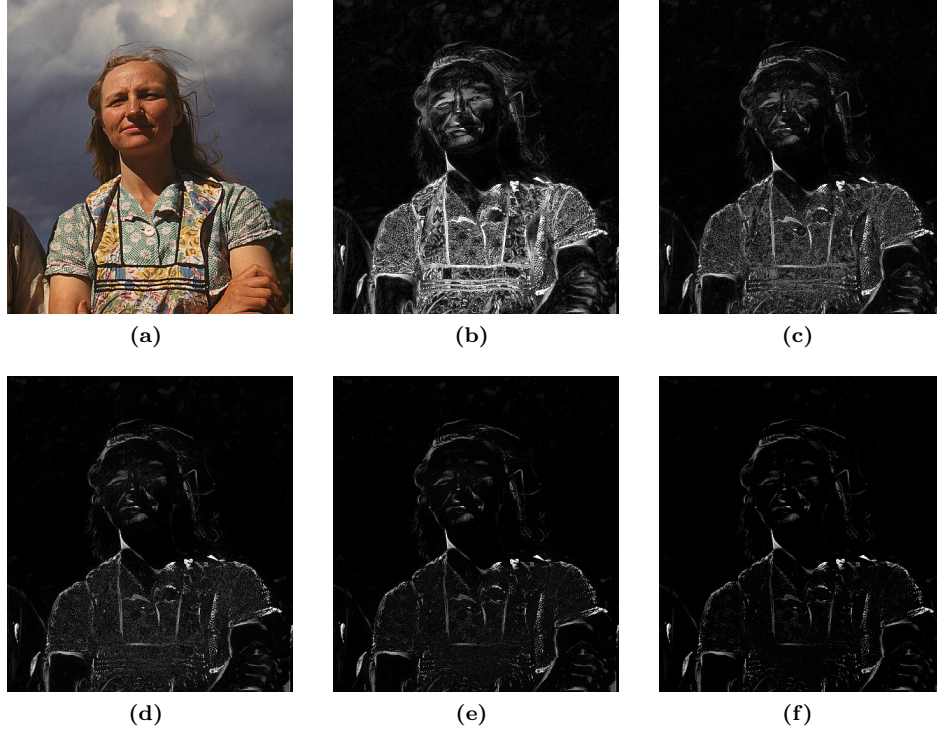


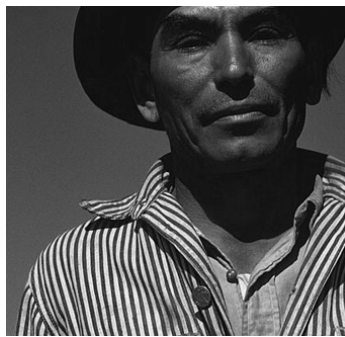
Figure 3.10: **3.10a** Original image $I_0 = I$. **3.10b** One iteration of corner map $I_1 = |I - M(I_0)|$. **3.10c** I_2 . **3.10d** I_3 . **3.10e** I_4 . **3.10f** $I_5 \approx \text{Corner}(I)$.

are given a low weight at one stage, they should also be given a low weight at every subsequent stage. Therefore, for each scale after the first and coarsest, we multiply each edge map produced so far.

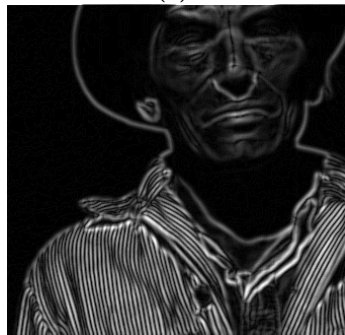
$$F'_S(I) = \prod_s^S F''_s(I) \quad (3.9)$$

Finally, we would like to include in this map some sensitivity to the distance of any pixel from important edges; preferably, pixels in empty regions of the image, far from edges and other salient features should also have less weight. In chapter 7, we see that this encourages the refinement of salient areas of the target image and discourages refinement in empty areas. We expand the edge map F' by some radius R representing the falloff distance of the weight map. To do this, we take a normalized Gaussian filter of the edge map and then multiply it with the original edge map F' . We also enforce some minimum weight c for empty regions by adding it as a constant, since we prefer that all errors be accounted for to some minimal degree, no matter where in the image they lie. That is,

$$F_S(I) = G_R(F'_S(I)) \cdot F'_S(I)(1 - c) + c \quad (3.10)$$



(a)



(b)



(c)

Figure 3.11: An improved image edge map. **3.11a** Original image. **3.11b** Gradient Magnitude. **3.11c** Corner map of gradient magnitude, F'' .

CHAPTER 4

MATCHING

4.1 Introduction

As you recall, in chapter 1 we described the six main processes that constitute the photocollage system: preprocessing and construction of necessary feature maps; the partition of the target image into a set of tiles; the matching of each tile with a portion of some library image; the composition of all images into a collage; color correction; and error calculation for further refinement passes.

In this chapter we first discuss the process for partitioning the target image into a set of tiles, and then the process for matching each of these tiles with an image from the image library. Along the way, we will discuss the aesthetic and efficiency considerations that go into the development of each procedure. We then outline a process for recalculating regions that were initially calculated unsatisfactorily.

First, we will begin a more detailed examination of the requirements for our collage in the areas of accuracy and discernibility.

4.1.1 Achieving Accuracy and Discernibility

In chapter 1, we defined the ideas of accuracy and discernibility, using the work of Orchard and Kaplan [69] as a reference. Often, accuracy and discernibility are treated as global properties of an entire collage, but really there is no requirement that accuracy or discernibility be uniform across the entire collage. Indeed, there are regions within a single target image in which high accuracy is required and other regions in which high accuracy is *not* required. We may scale discernibility proportionally as a function of the level of required accuracy. So, in stochastic, empty, and otherwise uninteresting areas of the image, we may increase our efforts towards discernibility while relaxing our requirements for accuracy. By localizing the priority between accuracy and discernibility we may mutually satisfy our requirements for maximizing both of these qualities.

Two characteristics, of relevance in this section, that are tied to accuracy and discernibility are: absolute size of the library image used; and the proportion of the library image contained within the tile—particularly the proportion of image objects or other interesting image features contained within the tile.

It is easier to discern the content of images that are reproduced at a larger scale. As for the proportion of the image used within the tile, it is also clear that images fragments containing more of the complete image are easier to interpret by a viewer, since more context is available in which to interpret image objects and structures. Furthermore, we assume that image regions in which objects are reproduced completely are preferable. See figure 4.1 for an example of various different styles of image fragment extraction, and their effects on discernibility.

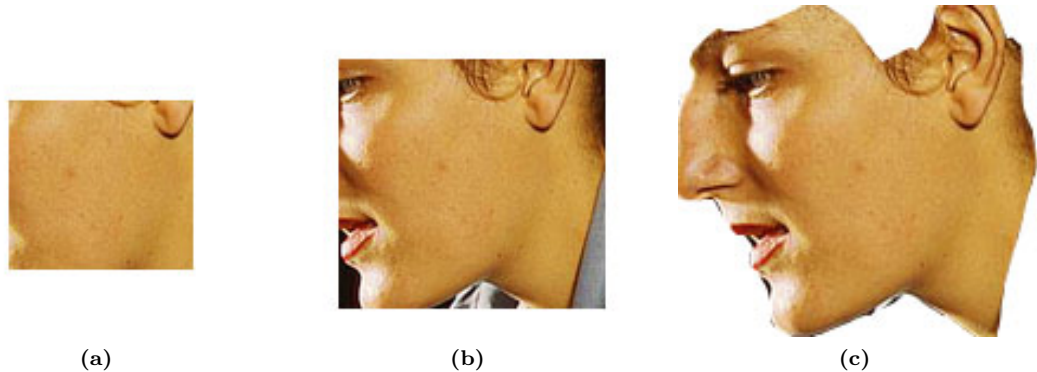


Figure 4.1: **4.1a** A small portion of library image, composited into a square tile. **4.1b** A larger portion of the library image, composited into a square tile. **4.1c** Composition with the tile boundary carved around image features.

How does the size and completeness of image tiles affect accuracy? Supposing we are drawing tile images from a fixed, finite database, Tran et al. [92] have found that accuracy increases as tile size decreases. This makes sense, since with small tiles the mosaic quality of aggregated elements is more responsible for the depiction of the target image than the contents of the tile images themselves. So, what we see is that while large tiles and near-complete image fragments are desirable for discernibility, they are, under normal image mosaic circumstances, detrimental to accuracy. However, as we have said, accuracy requirements may be localized within the image, so we needn't constrain the tile size uniformly across the entire image. Interestingly, even the difficulty of meeting more stringent accuracy requirements is locally variable across the image. What we mean by this is that even in a location where high accuracy is necessary, we may easily find an accurate match through a fast, somewhat cursory matching process; however, this point comes into play in match acceleration more than it does in accuracy/discernibility balancing.

The method of Orchard and Kaplan spends a large amount of time on matching details; the details of the target image are matched with details within the library images. Medium scale image features and object data are matched in exactly the same way, and account for a proportionally equal amount of computational effort. For structures in the target image that are larger than the tiles themselves, no explicit matching occurs at all. These large scale features are captured by the mosaic quality of the product image. That is, at this scale, the representation of these features is

carried out by the aggregation of atomic elements. In many tiles the time spent matching details to details is wasted, since many tiles will not contain significant detail in the target image. This is somewhat mitigated by signature matching techniques such as Di Blasi et al. [10], Zhang [99] and Range and Finkelstein [36]. We wish to spend more effort matching large-scale features and less effort matching non-salient small-scale details. We wish to, in essence, eliminate the mosaic characteristic as much as we can by choosing tiles that contain object structures, and matching these structures entirely through the content chosen for each tile.

In summary, we may relax the constraints of accuracy in certain special cases in order to allow a more discernible result. We will discuss the practical, implementation consequences of these observations in more detail in this chapter.



(a)

Figure 4.2: “The Great Paranoiac” [84] by Salvadore Dali. This piece exhibits both high accuracy and high discernibility. Furthermore, the matching between the target image, the head of a man, and the element images, human figures engaged in varying poses is performed on a large scale; this image does not rely at all on mosaic properties to represent the target image.

4.2 Collage Tiles

We wish to find a tiling for which there is likely to be a set of library images that fulfill the criteria we have established for a satisfactory photocollage. We wish to find a tiling for which there is a chance for large-scale correspondences to be found between objects in the target image and objects within the database images. At the same time, we wish to find a tiling for which it is possible, in a reasonable amount of time and with a reasonably sized image database, to establish a collage with significant visual similarity between itself and the target image.

For this purpose, we want to arrange tiles so that important edges and important semantic data is contained therein. In doing this, we are shifting most of the responsibility for producing good matches to the matching algorithm, since tile edges will never or seldom line up with image features.



Figure 4.3: A detailed view of an image mosaic produced by Orchard and Kaplan [69]. In the red circle, the edge of the tile and the object boundary in the target image match. We wish for our own method to avoid this circumstance. **4.3a** Detail of target image. **4.3b** Image-mosaic.

Figure 4.3 shows a detailed section of a mosaic by Orchard and Kaplan in which tile/feature alignment occurs. One can imagine a tiling scheme that uses regions of relatively constant color in the target image (figure 4.4); in this case, much of the work in matching the collage data to the constraint image has already been taken care of as part of the tile calculation process; the matching algorithm then needs only to find image regions that have constant color similar to that of the region from which the tile was derived. While this effect can be quite striking, it is not quite consistent with the aims of our project.

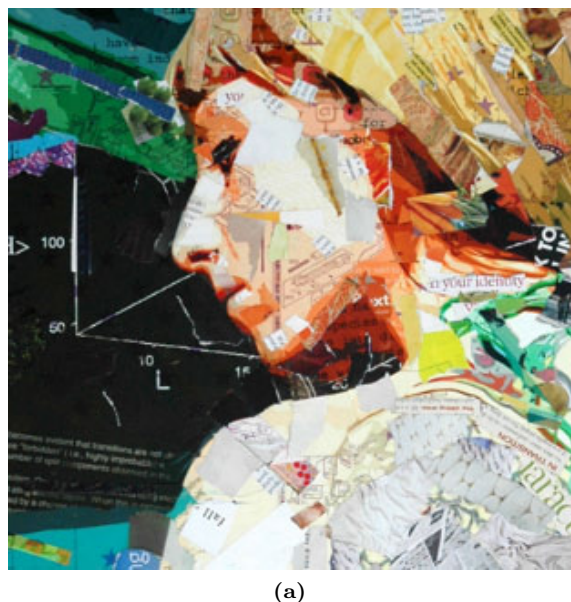


Figure 4.4: Detail of the collage “Brooke”, by Derek Gores [40]. For this portion of the collage, it appears that the artist has segmented the target image by color.

In some sense, this is the tiling for which the subsequent matching problem is the easiest. Conversely, the tiling mechanism we have proposed will produce a tiling for which the matching problem is more “difficult”, but we have chosen it because we propose that it will produce interesting results (in chapter 7 we show that this is true). The segmentation based method heretofore described solves much of the matching problem in advance by using the data present in the target image to produce a tiling that in itself captures much of the essence of the image. By forcing a tiling that is out of sync with image data, as we propose, we are forcing the matching algorithm to find a result that derives all of its essence from correspondences between data in the database images and the constraint image. Edges and features of the target image must be duplicated through the content of the tile images alone.

Although the tiling we propose has made the matching problem more difficult from the outset, we have devised the matching algorithm so as to partially alleviate this difficulty. This will be discussed in more detail in the following section.

We have explored two approaches to the problem of tile arrangement. Since we are referring to the original image content in order to find a satisfactory segmentation, we may still use ideas from the image segmentation literature for inspiration. However, the image segmentation literature primarily focuses on partitioning the image along important semantic content boundaries in the image, particularly object edges. Since we wish to find a segmentation for which edges are contained within the interior of regions, and not on region boundaries, we cannot use any existing segmentation algorithms that we know of. However, we may take a segmentation produced from an existing algorithm and use it as input for a process that produces a segmentation with the desired properties.

We have produced tilings with some desirable qualities by postprocessing segmentations generated via the mean shift method (MS), as implemented in the EDISON segmentation software [38, 63, 17, 16]. In order to produce the regions for our segmentation, we essentially adapt the technique for producing Voronoi tessellations to the task of processing edges, rather than single points. An outline of the algorithm follows.

For each pair of regions in the MS segmentation that share any number of boundary points, we produce one region within the new segmentation. The new region is the union of the two old regions. Since each old region will be added to more than one new region, and we require our new segmentation to be a valid partition of the image (in which each pixel belongs to only one region), we must divide each old region so that some connected part of it belongs to each of the new regions, in a sensible manner. We do this by ensuring that each pixel of the old region will become a member of the new region for which it has the least distance from the closest pixel in the pair boundary.

Another way to describe this is to say that the boundary between each pair of regions corresponds to a new region, and the pixels that are contained within the new region are those which are closer

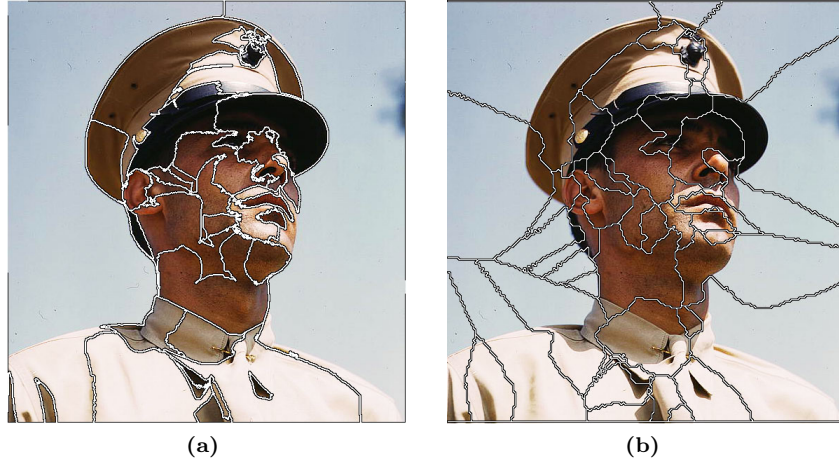


Figure 4.5: **4.5a** Mean-shift segmentation, with hand-chosen parameters, via EDISON [38]. **4.5b** Segmentation induced from the boundaries of the EDISON mean-shift segmentation.

to the pair boundary than to any other pair boundary. The results of this algorithm are displayed in figure 4.5.

We have found that it is somewhat difficult to control mean shift segmentation for the application of tile generation. It is difficult to choose parameters that produce tiles of consistently reasonable shape and size; in particular, it is difficult to choose a set of parameters for the EDISON segmentation system that produce desirable results for most input images. We have found that hand-tweaking is often required.



Figure 4.6: Three mean shift segmentations, produced by EDISON [38] with three sets of parameters. Note that for all segmentations the tiles vary widely in size, shape, and position.

The method of Voronoi tessellation is easy to control and creates regions that are simple, convex and compact. Also, Voronoi tessellations have commonly been used to partition images for mosaics [35, 43, 25]. We therefore have adapted this method for our own purposes. We have found that computing a centroidal Voronoi Tessellation, with randomized start points, produces adequate results for our purposes (4.7). A centroidal Voronoi diagram is a Voronoi diagram for

which the centroid of each region is positioned at the center of mass within the region for some mass distribution function [28]. For the mass function, we use a feature map computed from the constraint image. We compute the centroidal Voronoi diagram by Lloyd’s algorithm [27]. For the feature map, we use a modified gradient magnitude map (in which edges due to texture are removed), computed for a scale that depends on the size of the desired Voronoi regions. See chapter 3 for a more detailed discussion of the *corner map* that is used in this feature map calculation.

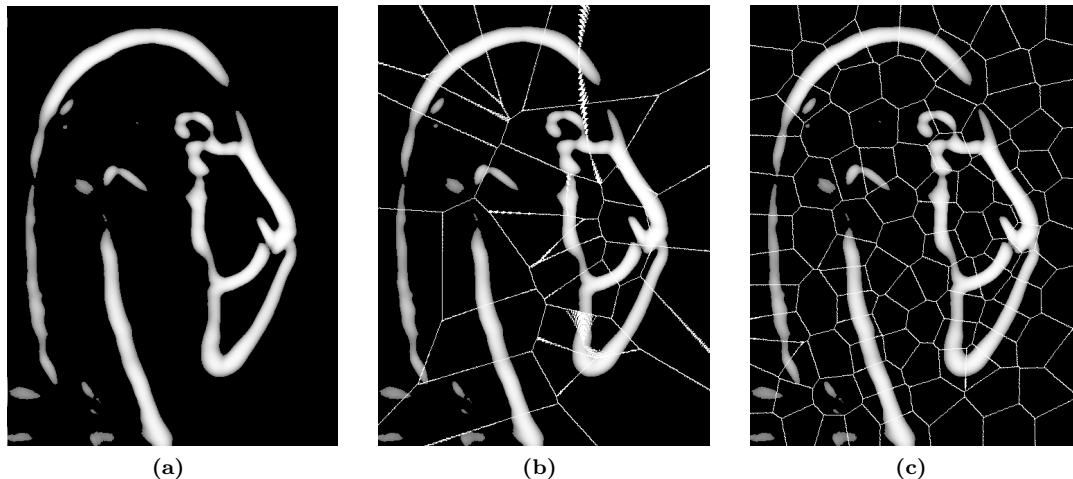


Figure 4.7: **4.7a** Feature/edge map. **4.7b** Voronoi tiling, with centroids placed randomly, but distributed according to the feature map. **4.7c** Centroidal Voronoi tiling, using the feature map as the mass function, with initial centroids placed randomly.

One problem of the Voronoi tessellation is that, aside from the placement of the centroids, the region shapes do not respond at all to the underlying image data. This may be solved by using bilateral Voronoi tessellation, with additional dimensions for additional feature maps. Bilateral Voronoi diagrams have been used for image segmentation by Inoue and Urahama [49]. However, as we have said, our main aim is to encompass salient image features inside tiles, and we have achieved this. Another effect of using a Voronoi tessellation, is that the area of each tile is relatively uniform, and a lattice which tends to the hexagonal in low mass regions. Little tweaking of parameters is required to obtain satisfactory results.

4.3 Matching Problem Formalization

For each tile, we wish to find an image within the database, and a sub-image within that image, that best represents the region of the target image contained within the tile. Previous investigators [52, 36, 69, 51, 9] have used SSD or sum of absolute error metrics to measure perceptual distance between two images, by summing the distances between individual pixels. Ashikhmin [2] has indicated that simple *smooth* metrics for measuring image perceptual distance lead to over-smooth results when

used for texture synthesis, since object edges and corners are not properly accounted for. Therefore, we do not necessarily want to find the library image with the least absolute error, although this is a factor in the calculation. We wish to consider higher order image features in order to more easily find a match that is appropriate in the context of the collage. More specifically, we compare each sub image in the database with the current sub image of the target image according to the following criteria:

- Step edges in the constraint image should be matched to step edges in the database image. Edges should match on a large scale (relative to the level of coarseness being evaluated presently), and minor deviations in the edges which do not correspond to each other should not unduly affect the score.
- The chosen sub-image should have low total absolute error wherever possible, except where it contradicts the other criteria.
- The chosen sub-image should have similar textural and smoothness properties to the region of the original image being examined, particularly texture contrast. Texture contrast is used as a key feature in many image segmentation systems [59].

We will compose an energy function that contains a term for each of these criteria, and then find the best collage according to these criteria by minimizing the energy function. The energy E is defined as

$$E = w_l E_l + w_c E_c + w_t E_t. \quad (4.1)$$

where E_l is the luminance term, E_c is the chrominance term, E_t is the texture contrast term. The quantities w_l , w_c , and w_t are parameters that control the influence of each term in the function. These parameters may be chosen by the user in order to favor particular quality factors. In order to reduce the number of parameters, one may vary the chrominance parameter proportionally with the luminance parameter, as such: $w_l = 1 - w_c$. In chapter 7 we demonstrate the effect of varying the components of $\vec{w} = (w_l, w_c, w_t)$.

4.3.1 Calculation of Energy Terms

All calculations are performed in YIQ color space, so that luminance and chrominance may be separately treated. It is known that human perception is more sensitive to luminance information [90], so it is preferable to treat it separately from chrominance. See chapter 7 for further discussion on the separation of luminance and chrominance. We then have, for an image I ,

$$I = (I_{y'}, I_{i'}, I_{q'}). \quad (4.2)$$

Luminance Energy, E_l is calculated as a simple sum of absolute differences of luminance values across the two images being compared:

$$E_l = \sum_{(x,y) \in A} F(x,y)M(x,y)|I_{y'}(x,y) - T_{y'}(x,y)| \quad (4.3)$$

$A = I \cap T$ indicates the rectangular region of intersection between the library image I and target image T , and $M \in [0, 1]$ is a real-valued function representing the tile mask. $M(x_0, y_0) = 0$ indicates that the pixel at (x_0, y_0) is outside the current tile (we set $M = 1$ for all $(x, y) \in A$ in order to ensure that the resulting match is valid even with alternate tile boundaries—this is not required, however). $F \in [0, 1]$ is a real-valued map that indicates the strength of important edges within the target image. $F(x_0, y_0) = 0$ indicates that the pixel at (x_0, y_0) belongs to an important edge. The rational for including this factor in the sum is provided in the section 4.3.2, which follows. Further details of the calculation of the map F are provided in chapter 3.

Chrominance Energy, E_c is calculated in the same way as E_l is, but the sums of each color-component are combined.

$$E_c = \sum_{(x,y) \in A} (F(x,y)M(x,y)|I_{i'}(x,y) - T_{i'}(x,y)| + F(x,y)M(x,y)|I_{q'}(x,y) - T_{q'}(x,y)|) \quad (4.4)$$

Texture Contrast Energy, E_t is calculated as the sum of absolute differences between the texture contrast values across the two images. The magnitude of each texture contrast value I_τ is proportional to the contrast of the texture at the pixel (x, y) . Details of the calculation of this map may be found in chapter 3.

$$E_t = \sum_{(x,y) \in A} F(x,y)M(x,y)|I_\tau(x,y) - T_\tau(x,y)| \quad (4.5)$$

4.3.2 Details of Energy Term Calculations

A Note on Texture Contrast as an Image Feature One characteristic of image mosaics we have observed is that they often contain high-amplitude noise, relatively uniformly spread over the area of the mosaic image, regardless of the noise and texture properties of the original constraint image. In regions of the constraint image that contain a high amount of noise or active textures, this “collage noise” may not significantly degrade the visual effect. However, in regions of the constraint image which are smooth, this decidedly unsmooth activity in the mosaic can be very distracting, as in figure 4.8.

Therefore, it may be desirable, for certain images, to match smooth areas of the constraint image with smooth regions of the library images. Texture contrast is, in some sense, a measure of



Figure 4.8: There is a high amount of noise in the image mosaic, caused not only by tile boundaries, but by high-frequency structure in the tile images. **4.8a** Detail of target image. **4.8b** Detail of mosaic.

smoothness. Our texture contrast map is preferable to a simple gradient magnitude map because it is not unduly influenced by non-texture edge data, and it does not suffer from haloing artifacts (spurious light and dark gradients abutting edges), as some other smoothness or texture contrast maps do. For these reasons, we have designed our energy minimization framework with texture contrast in mind.

Edge Feature Map F

In the previous sections, we used an edge map F to weight the influence of differences between pixels in the library and target images. Now we will briefly explain the calculation of this edge map and discuss its uses.

Before the match calculation process begins, a feature map $F(I)$ of the target image I is calculated for several scales, from coarse to fine. The scale of F that is used is roughly proportional to the size of the tiles being analyzed; for large tiles, a coarse scale is used, so that F represents large-scale, high contrast edges. For smaller tiles, F contains finer, more delicate edges, as well as more coarse edges. We obtain edge magnitude data for F that is unbiased by edges belonging to texture features, since we treat texture data separately from edges. For specific details on the calculation of F , please see chapter 3, section 3.7.

The absolute error is then multiplied with an edge calculation for large scale, high contrast edges. The purpose of this is to suppress the influence of errors in the collage that correspond to these prominent edges. We do this in order to improve the discernibility of the matches in the neighborhood of this type of edge; if we did not, the collage might be over-refined in these areas, obscuring the content of the collage images.

A Note on Large-Scale, High-Contrast Edges It may seem counterintuitive to use a relaxed match criteria in regions of the image where our feature map has responded to high salience; in this case, our decision is a matter of scale. The feature map we have computed responds to high

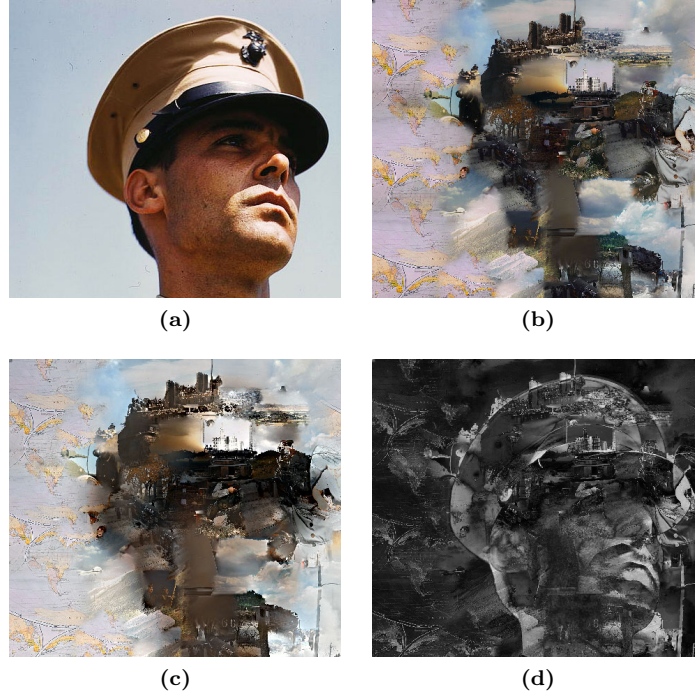


Figure 4.9: **4.9a** Region of the target image containing a high-contrast step edge. **4.9b** Library image composition after matching. **4.9c** Library image composition after matching and color correction. **4.9d** Error between target image and collage image.

contrast, large scale edges; that is, edges which have a high step, are relatively long, and relatively unbothered by deviations and curves from the next higher scale. In essence, these edges are simple and bold, and can suffer a great deal of degradation before they no longer are able to convey the information to which they had been originally charged in the target image. We hypothesize that these regions, because they can withstand degradation, are perfect candidates to take large, uninterrupted images patches from the library, regardless of high errors between the library image and source image that might occur. Figure 4.9 shows an example of the results of this strategy.

Image Scaling

Each image, as it is compared to the contents of the current tile, is scaled so that its area is proportional to the area of the current tile. This is done for two reasons; first, it is assumed that the size of the content of the library image is roughly scaled to the size of the image in which it is contained (we rely on the user to ensure this); therefore, to faithfully represent the content of the database image, we must ensure that its sub-image is proportional to the size of the overall image. We do not wish to take a portion of the database image that appears as nonsense when composited, out of context, with the entire collage. The easiest way to ensure this is to use the entire image, rather than a portion. Unfortunately, as Orchard and Kaplan observed, the lost degrees of freedom

for shifting the image into an optimal position will result in a poorer match. Rother et al. [78] trimmed the library images to a region of interest, rather than using the entire image; while this method has its uses, which they describe, and certainly ensures that important image information is preserved within the collage, it does not help with the problem of increasing match efficacy. As always, we wish to find a balance between discernibility and accuracy.

A second reason for scaling the image to be proportional to the tile area is to reduce the number of possible shifts and thus speed up the match calculation. Again, we must balance speed considerations with quality considerations.

Choice of Region of Interest We want to choose a subregion that has both represents a large portion of the main content of the image, but also closely matches the content of the target image within the tile. We prescale the image so that its largest dimension is proportional to the longest dimension of the tile, and we do not search scale space, so we are actually only computing the match score for a shift in two dimensions.

However, we would still like to preserve as much semantic content from each library image as possible. In particular, we should avoid, whenever possible, having an important object in an image only partially reproduced in the collage (although this is difficult in practice). We may avoid some of these cases by sculpting the boundaries between images such that the objects within images are considered and not arbitrarily cut off. This problem is similar to and, in some cases, equivalent to the seam repair process undertaken in patch based texture synthesis techniques and collaging techniques, as described in chapter 2.

Rother et al. [78] use a *region of interest* calculation that favors the center of the image as well as regions with high entropy, where entropy is a measure of the uncertainty of the value of each pixel, in a statistical sense [80]. Each image is preframed, before the composition of the collage, so that 90% of the image entropy is contained within the region of interest. Since our tile partition is not directly influenced by the distribution of information in the target image, and since the entropy of the portion of the target image contained within a given tile is arbitrary, this scheme is not entirely suitable for our purposes. For instance, it may happen that the target image within a tile contains very little entropy, and thus perhaps the portion of the library image chosen for that tile should also contain very little entropy. We have chosen only to favor central regions for our region of interest calculation, and control the utilized proportion of the library image as a parameter. That is, shifts of the image, for the purpose of the match calculation, are performed with respect to the center of the image. We are faced with the task of satisfying two goals, which may, at times, conflict with one another. We wish to use interesting regions of each library image, in service of producing a collage with high discernibility, but we also wish to maintain high accuracy in representing the content of the original target image. These aims will conflict when the region of the library image that

allows for the most accuracy contains little of the interesting content of the database image. We can exercise some level of control over this trade-off through a scale parameter s_{ROI} , which allows us to control what proportion of the library image is compared, and thus the number of possible shift positions. Using a larger scale parameter will result in the availability of a larger number of shifts. Then, where \mathcal{D}_T is the longest dimension of the current tile, the corresponding dimension of the current library image \mathcal{D}_I is recalculated as follows:

$$\mathcal{D}_I = s_{\text{ROI}} \cdot \mathcal{D}_T, \quad (4.6)$$

and the other dimension of the library image is appropriately scaled to maintain the original image proportions. We allow the library image only to be shrunk, and not inflated. Since the library image should never be smaller than the current tile, we have $s_{\text{ROI}} \geq 1$.

A Note on the Effect of the Scale Parameter s_{ROI} Lower values of s_{ROI} favor the discernibility of the library image within the tile, because they allow a large proportion of the library image to be used; discernibility is high because we are discarding very little of the image, and most discarded data will lay on the edges of the image. However, in these cases there are fewer shifts available for comparison, and thus the only chance for high accuracy is for the entire image to match the contents of the source image within the tile; this is unlikely for a general image library. Conversely, values of s_{ROI} that most favor accuracy in the collage are those that use only a very small portion of the library image. In this case, there are a high number of shifts available, and thus the chances of a very good match are relatively high. However, the chances that this small portion of the image will contain a high amount of interesting information is relatively low, for a general, non-stationary image. So, in figure 4.10 we see that s_{ROI} controls, to a large extent, the balance between discernibility and accuracy on a global scale.

Alternately, we may adjust s_{ROI} as a function of refinement level or tile size in order to add some localization to these scale effects. We explore this further in chapter 7.

4.4 Acceleration of the Matching Process

Traditionally, a large library of images has been required in order to produce an image mosaic of good quality. This is due to the fact that mosaic quality appears to be linearly correlated to library size [92]. Therefore, it is desirable to have a large library to search, which leads to a lengthy search process if performed by brute force. Ideally, we would like the user to be able to compose a collage from a library of images that is arbitrarily large, without an extended wait time. It is necessary to have a method of searching a large library of images fast enough that a collage may be computed in a reasonable amount of time (the definition of *reasonable*, in this case, being somewhat dependent

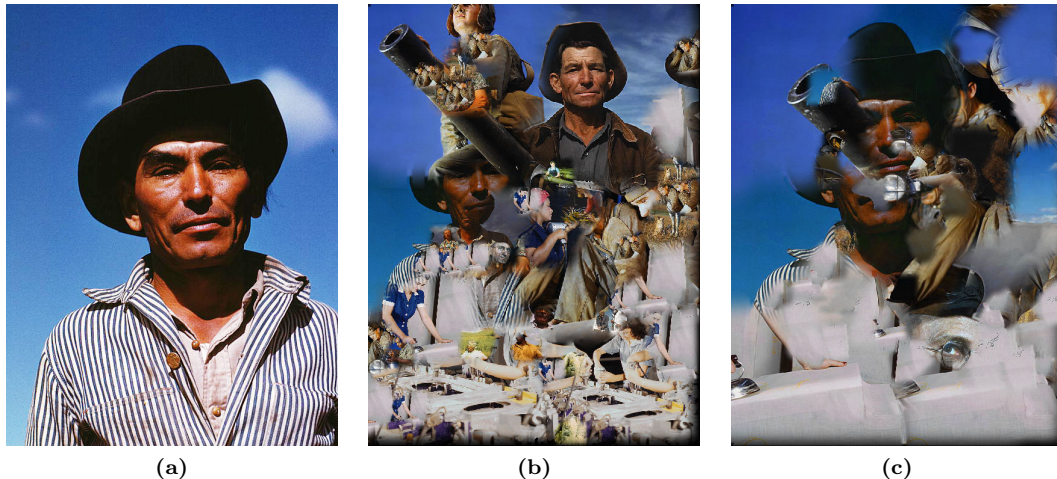


Figure 4.10: **4.10a** Target image. **4.10b** Composition with low s_{ROI} **4.10c** Composition with high s_{ROI} While the discernibility of the composition in 4.10b is higher, it is apparent that the composition in 4.10c is a closer match to the original, 4.10a

on how we define *reasonable quality* for the output).

Naïvely done, matching takes much too long. Using brute force, we would have to compare each pixel in each sub-image of each database image to each pixel in each sub-image of the target image. The time complexity of this operation is $O(KM^2N)$ where K is the number of images in the library, M is the number of pixels in each library image, and N is the number of tiles in the constraint image. We have M^2 rather than M because we must measure each of M pixels in a given library image for each of M possible spatial translations with respect to the target image (that is, for which the library image is entirely contained within the target image, and for which the tile region is entirely contained within the library image). Figure 4.11 shows a typical setup for a match calculation. We have adopted acceleration techniques that do not reduce the time complexity, but nevertheless reduce the computation time to within feasible limits.

In the photomosaic literature there have been two main approaches to match acceleration. Coming at the problem from the direction of increasing the speed of matching at the cost of accuracy, several approaches have been developed that rely on the precomputation of low dimensionality signatures which may be quickly searched [10, 50, 36, 99]. From the point of view of improving accuracy while attempting to maintain reasonable efficiency at the cost of speed, Orchard and Kaplan [69] have advocated the online computation of match scores between image sub-regions, taking into account various shifts both in image and color space. We use a method that combines these two approaches into a matching mechanism that can quickly search through a large image collection, screen out the poorest candidate images, and then switch to a more accurate method.

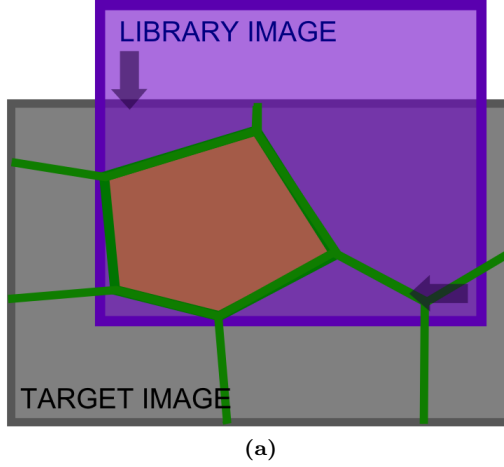


Figure 4.11: For each tile, outlined here in green, each pixel of the library image, shaded purple here, must be compared with each pixel of the target image within the tile, shaded orange.

4.4.1 Hierarchy

We employ a hierarchical approach in which fast, low level calculations are used to pare down the database to a small subset of likely match candidates, in several stages. As the complexity of the feature set increases with each stage, the number of images whose features must be calculated and compared decreases. In this way, we are able to use a system of successive estimations to find a set of good match images, without having to exhaustively search the entire database. This is similar to the pyramidal coarse-to-fine image matching schemes used in image registration [101] such as those of Wong and Hall [98], Zheng and Chellappa [100] and Kumar et al. [56].

We begin with a fast initial matching pass, inspired by content based image classification methods such as those employed by Oliva et al. [67], which is intended to reduce a large library of images to a small set of good candidate images. We then perform a sequence of pixel-wise comparison passes, one for each of various levels of resolution, for all valid translational shifts of the image against the tile mask, reducing the library at each resolution level, so that fewer images are available for consideration at the next stage. At the final and highest resolution of consideration, the image subregion with the best match score is chosen for composition into the final collage.

First Matching Pass

The purpose of the first matching pass is not to find good image subregions for composition into the current tile, but to reduce the image library to a subset of images, each of which is likely (or more likely than the set mean) to contain a good image subregion. Therefore, we do not calculate a match calculation over all potential shifts of the image against the tile, but only calculate a coarse set of statistics, which we compare with the statistics of the source image in the current tile. Oliva

et al. [67] were able to analyze large scale features of images in order to categorize them according to semantic content. We wish to categorize the images in our library not by semantic content, but by the prevalence of similar regions to the tile region currently under consideration. The match calculation we perform may be summarized as the minimization of the following equation:

$$E^0 = w_c E_c^0 + w_t E_t^0, \quad (4.7)$$

where E_c^0 is the luminance and color term and E_t^0 is the texture-amplitude term. In this equation, E^0 is distinct from E in equation 4.1. The quantities w_c and w_t are parameters that control the influence of each term in the equation.

Calculation of Energy Terms We calculate the value of each term by computing the mean value of the target image, within the current tile, and comparing it with a histogram for each library image. For each of the luminance, chrominance and texture contrast terms, we measure the height of the bin, in each library image, that corresponds to the mean value within the current tile. An image with a large bin for a given value will ostensibly contain a higher proportion of image data corresponding to that value, and therefore has a higher chance of providing a good match. In this way, we wish to use the histogram of each library images to estimate, by bin height, its suitability as a match for the current tile.

For each image, a coarsely binned (eight bins) histogram is calculated. The histogram is composed of local mean luminance, chrominance, and texture contrast values, the mean being taken over a region with area close to the area of the current tile under consideration.

A score is then calculated for each database image. The score is proportional to the number of elements in the bins corresponding to the mean color and texture contrast of the image sub-region within the current tile. Specifically, we subtract the height of the bin from some sufficiently large positive integer, no less than the area of the current tile, so that the result is always positive.

The motivation for this procedure is that library images with a high number of pixels similar to the tile mean are likely to give good pixel-wise matches. This assumption will fail in some cases: for instance, when the local mean differs significantly from the actual image region. However, since image data is often correlated across scales [5], and images are understood by the human vision system according to a multiscale decomposition [68, 83], we believe this is a reasonable and effective scheme. In chapter 7 we show that this strategy does indeed accelerate matching with acceptable error costs. The images with the best matches are kept for the next pass, according to a proportion specified by the user.

Subsequent Matching Passes

Once a subset of desirable images has been gleaned from the library, a series of identical matching passes is performed with each pass utilizing a finer scale than the previous. In each of these passes, a full match calculation is performed; each pixel of the library image is compared with each pixel of the current tile, over every possible translation, as described in the previous section. The database images with the best matches are kept for the following pass, where the match calculation will be performed at a higher resolution, thus increasing the accuracy. The proportion of images kept for the next pass is, again, specified by the user. After the final pass, where a small number of images is analyzed at full or near full resolution, a single image subimage is selected for composition into the tile.

We have chosen to limit the maximum resolution for matching to $1/4$ of the full resolution, in order to maintain a high processing speed. Of course, it is possible that there exists somewhere in the library a near-perfect match at full resolution; we assume that this ideal case is unlikely for most libraries.

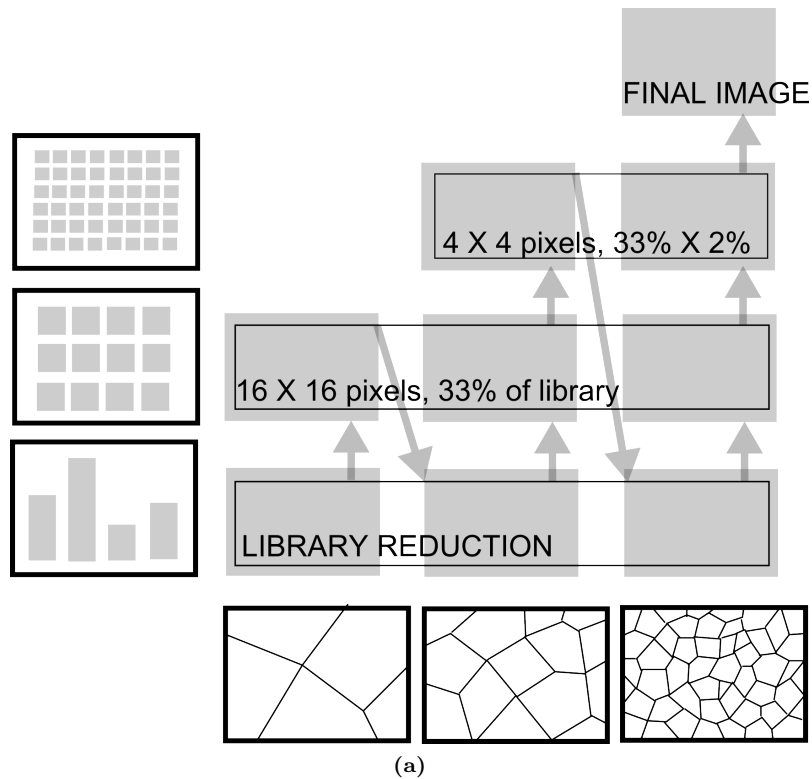


Figure 4.12: A visual guide to the matching process.

Preprocessing

In order to speed up the comparison of image features, we precalculate the features for each image in the database and save a new file for each library image and each scale level. This dramatically cuts down the time spent loading each image and extracting its features. Preprocessing needs to only be performed once for a given library.

4.5 Refinement

Because we wish to use to the largest portion of each library image as possible, both for speed and aesthetic considerations, it is desirable to use as coarse a partition as possible for our tiling with the fewest and largest tiles. However, it is unlikely that any collage exists that meets our accuracy criteria, described in chapter 1, for such a coarse tiling. Therefore, in this case, we would like to reprocess certain parts of the collage to improve their quality.



Figure 4.13: 4.13a Collage. 4.13b Original target image. 4.13c Error between collage and original. 4.13d Collage with recomputed regions.

Since we make an initial attempt to calculate a collage with tiles of some initial size, and it is difficult to efficiently recalculate another configuration of similarly sized tiles to produce a collage that shows any improvement, we compute a new tiling, which is less coarse and contains smaller tiles. In this way, we relax our initial conditions in order to simplify the problem and obtain better accuracy. We calculate matches for only those tiles in the new tiling that lie over undesirable regions of the last computed collage. The new image matches are then composited over top of the old collage. This process is iterated until a desirable result is obtained, as shown in figure 4.13.

We define “undesirable regions” of the last computed collage as regions in which the error density between the collage and the original image exceeds some threshold, or which fall into the top percentage of tiles in error density (we have chosen to refine the worst 25% of tiles, as explained in chapter 7). By “error density”, we mean the total sum of the pixel-wise difference between the current collage and the target image, divided by the area of the current tile. The pixel-wise difference between the current collage and the target image is expressed as the map \mathcal{E} . Also, since we require higher accuracy in some regions of the collage, and higher discernibility in others, we multiply the error \mathcal{E} map by a salience map S , so that high error neighborhoods are not refined if they do not contain significant data. For example, if high errors occur in a neighborhood that contains open sky in the original image, we may not wish to refine it any further; a looser match is sufficient here, and perhaps even desirable, since further refinement could break up and obfuscate the content of the library image that has been composited into the collage. Each refinement operation has a cost, which is paid in the discernibility of the collage. Therefore, we wish only to perform refinement where it is more important to transmit the data from the original target image than it is to faithfully preserve the data from the library image; this priority is observed where the salience of the target image is high, according to the salience map S that we define in chapter 3.

Also, since in our match calculation, described in the previous section, we have weighted the contribution of each pixel by an edge feature map F , we must also modify our error map accordingly. We do not wish to refine regions in which we have already agreed to tolerate a higher error threshold, so we therefore multiply our error map by the same edge map. Our final error map is thus defined as follows:

$$\mathcal{E}_{\text{final}} = \mathcal{E} \cdot S \cdot F \quad (4.8)$$

where $\mathcal{E} = |C - T|$ is the absolute error between the collage C and the target image T , S is the salience map, obtained as the corner map $\text{Corner}(T)$ of T , and F is an edge map, obtained according to the method described in the previous section.

An error density, based on the error map heretofore described, is then calculated for each tile in the next, more refined partition. If the error density within each tile exceeds a certain threshold, or is among the highest error densities, that tile is activated and a match is recalculated for it in

the next pass.

In the previous section, we discussed a method for relaxing the matching criteria in the neighborhood of important edges. Since the salience map tends to respond to edges in the image, its effects will thus be canceled out near certain important edges, since we combine the maps of these phenomena multiplicatively; that is to say, we have chosen to relax the strictness of the match calculation near certain edges, so the high-salience of these same edges is thus disregarded in the choice of which tiles must be re-calculated. For this reason, we choose a scale for the salience map that is relatively high, so even though large scale, high contrast edges are left alone in the refinement process, other salient data will appropriately be flagged for re-processing, as in figure 4.14.

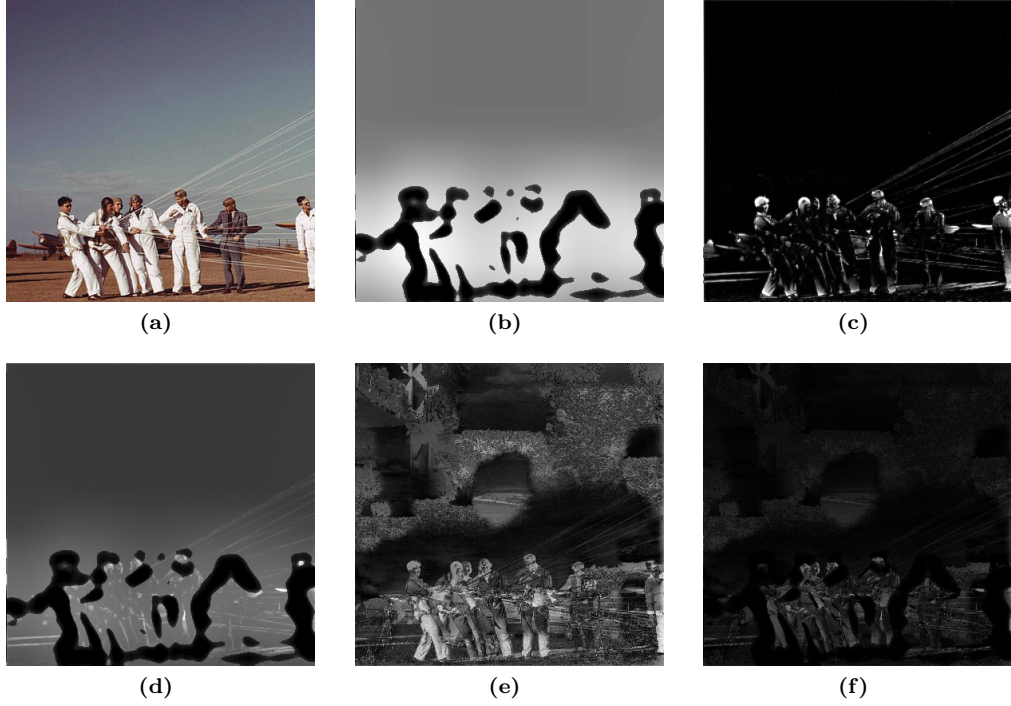


Figure 4.14: 4.14aTarget Image. 4.14a The edge map F , for increasing discernibility around edges. 4.14b The saliency map S , which indicates high priority for further refinement in future iterations. 4.14d The intersection of S and F . Note that each map has been calculated at a scale so that one does not completely contradict the other. 4.14b The raw error map between the collage and the target image. 4.14d The raw error map is multiplied with S and F to create the final error map.

CHAPTER 5

COMPOSITING

5.1 Overview

After the matching process, for a given refinement level, we have, at our disposal, the following data: a list of tiles along with their position coordinates; and a corresponding list of images, along with their coordinates and scales with respect to each tile. It is then our task to use these data to compose an actual collage image, which may be further processed or kept as the final result.

Compositing occurs after each collage refinement stage; the newly matched tiles are re-rendered and a new collage is produced. This compositing pass requires additional image analysis, and introduces additional information into the specification of the collage. This additional information, which consists of an edge map and error map of the collage so far, is used for the computation of the next refinement level. Ideally, all necessary information for the specification of the collage would be computed as part of the match calculation, also described in chapter 4. However, the image analysis and modification steps performed in compositing—specifically, color correction and seam-repair—are more costly in terms of computation time, and thus have been collected at the end of each refinement pass (for instance, it is impractical to compute a Voronoi tessellation for every possible library image at each tile). We wish to avoid the computation of a new tile partition and color correction for every possible image when only one single case will be used, in the end. This is the same philosophy that has guided our match-acceleration methods. In fact, the choice of tiles to be refined in the next iteration, which we described in chapter 4, is made *after* the compositing process, in order to take into account the new information that has been introduced into the collage through the compositing process.

5.2 Seam Repair

Efros and Freeman [29] use least cost paths through the error map between overlapping tiles to find visually inconspicuous seams. Rother et al. [78] use Poisson blending in the alpha channel, modulating the amount of feathering by edge strength.

Since we have used centroidal Voronoi regions to generate our tiles, we may reuse the Voronoi

centroids we have computed to recompute modified Voronoi regions that have the seam hiding characteristics we desire. This applies also for other types of Voronoi tessellations, and may even be extended to generalized region maps. However, the adaptation of this method for general regions may increase the computational complexity of the algorithm.

As in Autocollage, we desire image transitions that are softer in smooth regions of the image while more abrupt along image edges; also, as in Autocollage, we wish to respect semantic content within the image. In order to do this, we recompute our Voronoi tessellations in such a way that edge data from the library image is taken into account, with a provision for boundary edge feathering.

5.2.1 Path Cost Voronoi Diagrams

In order to recompute a Voronoi partition so that its boundaries adhere to important image edges, we may use a Voronoi diagram with path cost as the distance metric, as in Mould’s treatment of image-guided fractures [65]. Path cost refers to the cost of the least-cost path through a weighted graph, from each pixel to the centroid. The weights of the graph, as in Mould’s paper, are derived from edge information in the image to be composited, and the images that it overlaps. Where there is an edge between a region pixel and the centroid, the least cost path will not pass through the edge, but go around it; therefore, the cost of this path greater than it would be if the edge did not exist. The result is that region boundaries tend to fall along edges. The path cost for each pixel is calculated using Dijkstra’s algorithm [24], with each centroid placed in the heap as a start node. The use of Path Cost Voronoi diagrams offers several advantages for preservation of content within the tile images. Irregularly shaped image objects may be accurately carved out of the main image, effectively preserving semantic content and maintaining high discernibility. Figure 5.1 shows three intermediate stages in the course of Dijkstra’s algorithm.

Since we have in our graph one node for each pixel, the time complexity of this operation is $O(N \log N)$, where N is the number of pixels in the target image. We must process each of N pixels once, and there is an $O(\log N)$ cost associated with storing each node on the heap. Please see Mould’s paper [65] for details of the Path Cost Voronoi region method.

Edge Weights The edge weight map e , over which the path cost is computed, is composed as a combination of edge maps from each of the tile images being composited. First, we will describe the process for obtaining edges for each individual tile image, and then we will describe the process of combining them to form one overall edge map. All tiles are computed simultaneously from a single edge map that includes contributions from each tile.

For each tile image, a gradient magnitude map is taken for some scale coarser than the finest scale of the image, and then texture edges are removed. A Gaussian kernel is used to scale down the image. The texture edges, or edges due to textures rather than object boundaries, are removed

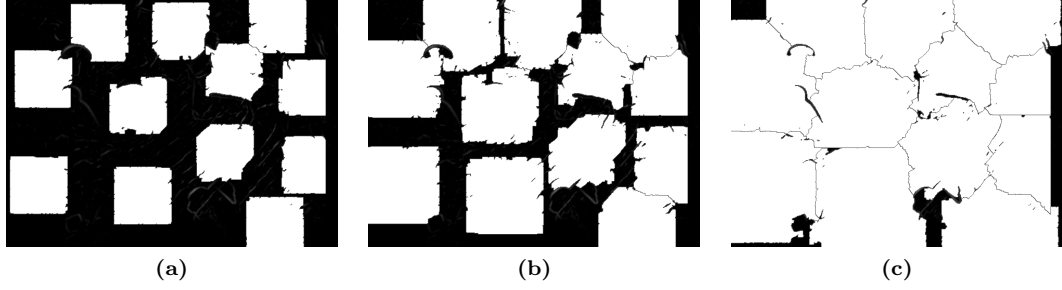


Figure 5.1: Three intermediate stages in the Path Cost Voronoi diagram calculation. The regions (white) grow larger with each iteration. The node weight map is shown in the background. Notice that regions curve around edges in the weight map.

via the corner map method; by taking the absolute difference of the image with its median filter, iteratively, we may suppress texture edges, as specified in chapter 3. We use only edges that do not belong to textures, because we wish to segment the image content by object, not texture; textured regions should not be treated separately. In essence, we are only interested in segmenting the images according to object boundaries, and thus wish to discount texture information altogether. We additively combine edges of several scales in order to capture larger features as well as certain prominent details.

In summary, for a given library image I , the contribution $e(I)$ to the final edge map e_{final} is as follows:

$$e(I) = \sum_{i=0}^M |\nabla G_{2^i} * I| \quad (5.1)$$

where M is the total number of scales used, ∇ is the two-dimensional gradient operator $\nabla I = [\partial I / \partial x, \partial I / \partial y]$, and $G_{2^i} * I$ is the convolution operation with a Gaussian kernel of width 2^i .

The purpose of recalculating the regions after matching is to resolve overlaps between tile images so that each collage pixel is assigned to only one image. As we mentioned before, we must update the partition of the target image to account for the library images that have been chosen to fill each tile. Therefore, for each pixel in the collage we must somehow take into account the edges of several overlapping images at once. We have done this by using, for each pixel (x, y) of the final edge map e_{final} , the maximum edge strength of all images overlapping (x, y) . Additionally, we weight the influence of each edge pixel according to its distance from the boundaries of the *original* Voronoi region associated with the current image. This distance weighting is calculated as a simple Gaussian convolution of a binary edge map of the original Voronoi region (using a Gaussian function of distance). After all, we wish for our new region boundaries to roughly follow the old boundaries, since it was these old boundaries that guided the matching calculations. The Gaussian filter simply produces a map that falls off in intensity, with distance, from the original segmentation boundaries. So, the final edge map e_{final} is defined as follows:

$$e_{\text{final}}(T, x, y) = \text{Max}(e_i(I_i(x, y)) \cdot v(x, y)_i), \quad (5.2)$$

for all edge maps $e_i = e(I_i)$ and where $v(x, y) \in [0, 1]$ is proportional to the distance from a boundary in the original Voronoi Tessellation, with greater weight closer to the boundary. we calculate v as follows:

$$v(x, y) = G_R(\text{TileBoundary}(x, y)), \quad (5.3)$$

where R is a scalar on the order of the distance across which tile images are blended, and G_R is the normalized Gaussian blur operation. Figure 5.2 shows various components of this edge map calculation.

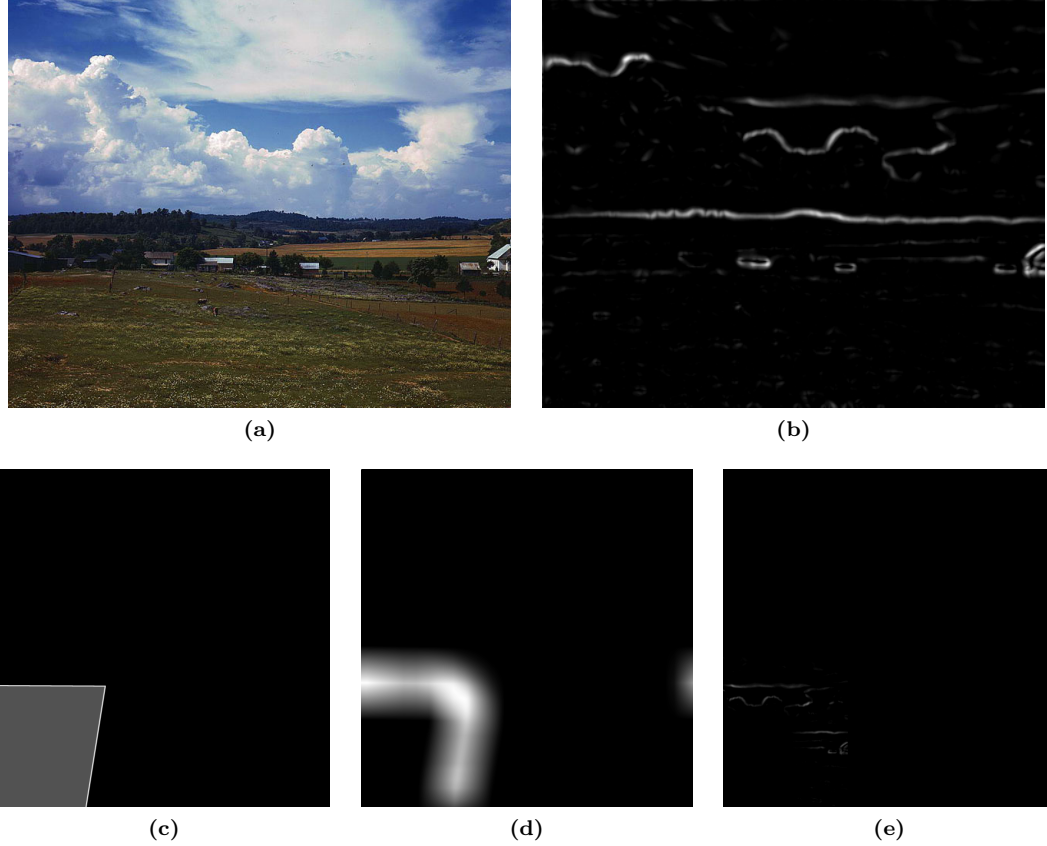


Figure 5.2: **5.2a** Original tile image I to be composited. **5.2b** Edge map e of I . **5.2c** Boundary of the tile to be composited, from the original Voronoi segmentation. **5.2d** Distance map v . **5.2e** Contribution of I to final edge map e_{final} .

Feathering To enable smooth transitions when necessary between collage tiles, we feather the tile boundaries by applying a simple adaptive Gaussian blur to the region masks, in order to convert each binary mask to a continuous mask with soft edges. Within the adaptive Gaussian blur

operation, the radius of the blur kernel is proportional for each pixel (x, y) to the strength of the edge in the map $e_{\text{final}}(T, x, y)$. The purpose of this is to allow wider, softer feathering in low-activity regions, while maintaining crisp region edges near object boundaries and other important edges in the tile images, as illustrated in figure 5.3.

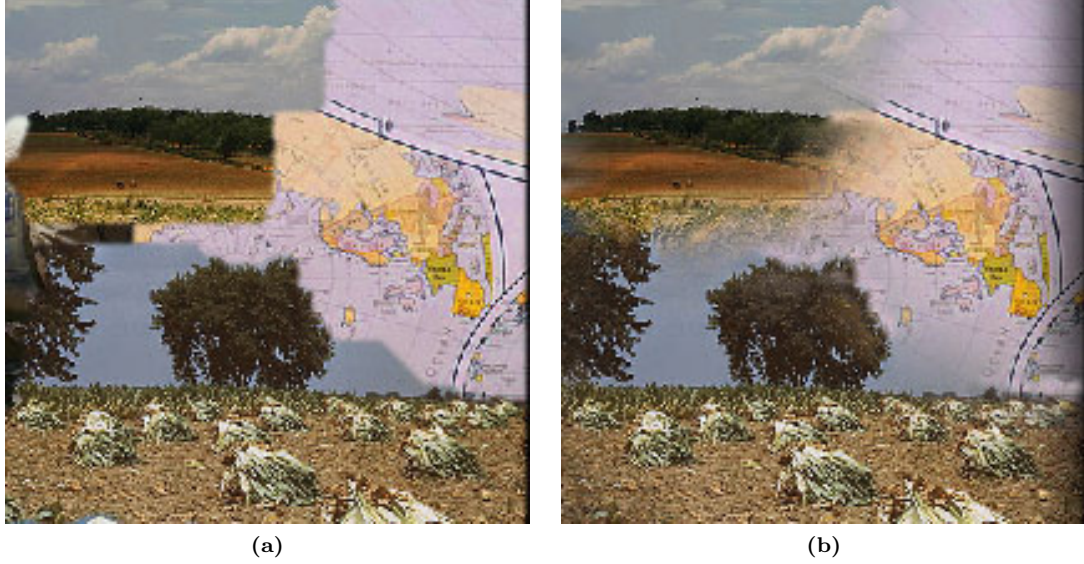


Figure 5.3: **5.3a** Composite with a narrow maximum feather radius. **5.3b** Composite with wider adaptive feathering. Notice that while tile boundaries near object edges, such as the tree, remain well defined.

CHAPTER 6

POST-PROCESSING

6.1 Motivation

It may happen that the best match for a particular tile resembles the content of the tile to a certain degree, but differs enough to cause a visual interruption in the transmission of the target image through to the collage. Similarly, it may be that the composition of the collage, even with seam repair and special transitions between images, results in distracting discontinuities between adjacent tile images that do not reflect the continuity of the original source image.

In these cases, we would like to alter each of the tile images, or the collage as a whole, so that they better represent the content from the target image. However, we do not wish to damage the content of the tile images themselves or impair their efficacy in representing the deeper level of content that is expected to be present in photocollages. In short, we do not wish to negatively affect the discernibility of the tile images.

Also, we wish to alter the images in such a subtle way as to respect the overall aesthetic of the mosaic composition process; we do not want to tamper with the final result in such a way as to cast doubt on the nature of the original collage creation process. With these considerations in mind, we have developed a postprocessing system for improving the visual appeal of the collage after it has been created.

Previous Approaches Postprocessing of image mosaics, in order to increase accuracy at the cost of discernibility, has been implemented in past systems. Finkelstein and Range [36] sought to match the average color of the target image within each tile by appropriately translating and scaling each color in the tile image. Orchard and Kaplan [69] pointed out that this technique often distorts the content of the tile images beyond recognition, therefore significantly reducing its discernibility. They proposed a technique that calculates the optimal shift for each color component, in YIQ space, as part of the match calculation. Figure 6.1 shows these two methods compared side by side. Di Blasi et al., in “Puzzle Image Mosaic” [9], also shift the overall color of each tile object so that it matches the mean color within the tile after the mosaic composition.

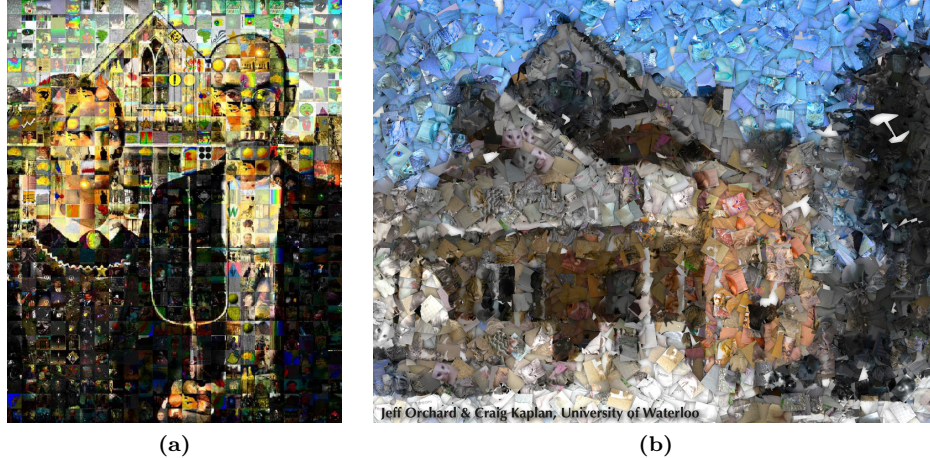


Figure 6.1: 6.1a The mosaic of Finkelstein and Range [36] has been color-corrected in a pixel-wise fashion so that the unedited content of the original image is clearly visible. 6.1b The mosaic of Orchard and Kaplan [69] has been color-corrected in a tile-wise fashion, avoiding the problems of the previous approach. However, accuracy in this case is dependent on the shape, orientation and size of tiles used.

6.2 Our Approach

As previously stated, our postprocessing operation should increase accuracy without negatively affecting the discernibility of the tile images within the collage. Simultaneously, we wish to avoid altering the collage in any way that allows image information from the target image to be reproduced directly in the collage, without any apparent attempt to use correspondences between images; any details of the target image which show through into the collage must be composed from an arrangement of details in the tile images, in a manner that appears to be natural. We have identical standards for the larger scale object and structures.

The method of Finkelstein and Range causes details of the target image to clearly show through in the mosaic. In some cases, natural correspondences between details in the two matched images, which on their own are quite clever and effective, are obscured by the color correction process. The method of Orchard and Kaplan does not suffer from this effect, since color shifts are applied uniformly across every pixel in each tile. In this case, the representation of details from the target image is handled entirely through the matching process. However, this global shift often produces tiles with strange color schemes, particularly when dramatic hue rotations occur (figure 6.2). Di Blasi et al. [9] and Kim et al. [51] perform color correction in an object-wise fashion, since objects are the primary rendering primitive in their methods.

We have combined aspects from all three of these approaches. Since we often use very large tiles, applying a uniform shift over each tile may not be appropriate; the desired level of accuracy in color matching may not be obtainable. Orchard and Kaplan were able to get away with this

because the tiles they use are small enough that this method has ample precision. We would rather apply corrections in an object-wise fashion, as do Di Blasi et al. and Kim et al., so as to preserve the content of the tile images and avoid the appearance of obvious tampering. However, since we compose the collage on the image level and not on the level of geometric shapes as Di Blasi et al. do, we require some basic object sensitive image filtering. The most obvious candidate for this operation is the bilateral family of filters, which we will discuss shortly.

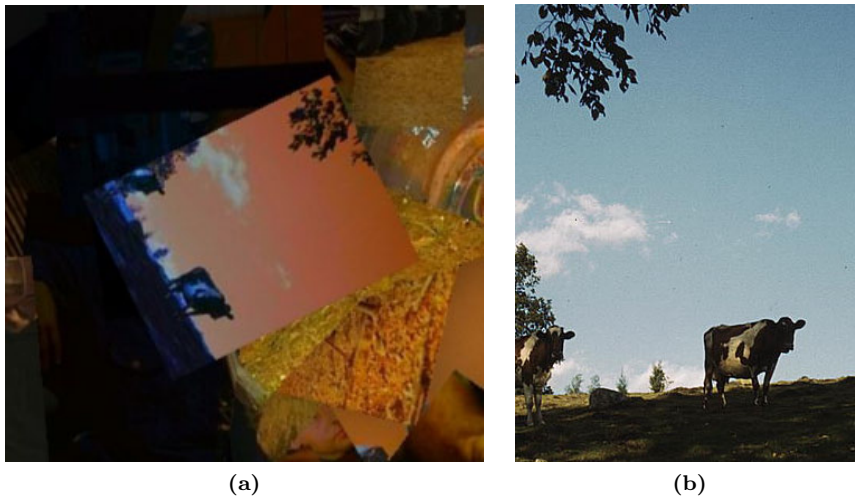


Figure 6.2: 6.2a The hue shifting scheme of Orchard and Kaplan sometimes produces odd-looking results, like these blue cows in a detailed section of the collage from figure 2.6. 6.2b The original library image.

6.2.1 Hybrid Images

Oliva et al. [68] have described a method for creating hybrid images; a hybrid image is an image that represents something different when viewed from far away than it does when viewed close up. The basic process for creating these images involves combining the high spatial frequency data of one image with the low spatial frequency data of another. Essentially, semantic content is segregated by frequency within a single image.

Hybrid images bear some similarity to image mosaics and collages in that different content is represented on different scales. Indeed, resolution of image features with respect to viewing distance has been used by Tran et al. [92] to test mosaic accuracy. However, there are some major differences; particularly, in the case of our project, we do not wish for there to be a hard boundary in frequency space or scale space between the database image content and the constraint image content. There should be a continuum across scales in which these two possibly disparate sets of information are represented.

Furthermore, the content of the target image in the collage should be somewhat visible even at

close range. This is not the case with hybrid images; see Oliva’s publications [83, 68] for a more complete discussing of correlation across scales in hybrid images.

Despite these differences, the idea behind hybrid images has motivated our approach to the post-processing of photocollages. While it is tempting to apply the hybrid-image method for this purpose, we will see, in the following section that a similar but more appropriate method may be devised, using an image decomposition scheme similar to the image denoising and detail transfer method of Petschnigg et al. [74] and Eisemann and Durand [31].

6.3 Method

One can imagine a process in which an image mosaic is artificially created by rendering a hybrid image between a general, non-representational collage (such as is produced by Autocollage) and some other base image. However, because there would, in this case, be no information correlation across scales, the resulting mosaic would not be convincing; the collage image may appear as a thin transparent layer overlaid upon the blurred target image (indeed, this is exactly what it is, as we see in figure 6.3).

One may achieve a better result by creating the base and detail layers in some manner other than simple spatial frequency high pass and low pass operations. Using a texture/object decomposition (or *base/detail*), such as that used in Fardbman et al. [33] and Bae et al. [3] for our base and detail layers, we are able to combine information from the target image and collage image in a way that is more faithful to the collage aesthetic, so that the target image is visible also at close range (figure 6.4, 6.5, 6.6). Precisely, for our collage C , we have

$$C = C_B + C_D, \quad (6.1)$$

where C_B is the base layer, or object layer, and C_D is the detail layer, or texture layer. Similarly, for our target image T , we have



Figure 6.3: **6.3a** Target image. **6.3b** Collage by Microsoft Autocollage 2008 [79]. **6.3c** Hybrid collage.



Figure 6.4: A modified hybrid image, with base layer created from a cross bilateral filter between the secondary image and the primary image, and the detail layer created from a bilateral high pass filter of the primary image. Some additional enhancement of the details from the secondary image have been done according to section 6.3.1. **6.4a** Primary image. **6.4b** Secondary image. **6.4c** Base layer. **6.4d** Detail layer. **6.4e** Final double image.

$$T = T_B + T_D. \quad (6.2)$$

However, simply using the object layer from the original source and texture layer from the collage, as in $T_B + C_D$, is not sufficient, since we wish to create the appearance that both objects and textures are being used from the collage; we wish to modify the contents of the object layer of the collage so that the *objects* within this layer, when viewed in aggregation, more closely resemble the *entire* target image. To achieve this, we must somehow adjust the object layer of the collage so that it more closely resembles the object layer of the target image.

First, we must find a way to decompose the collage into object and texture layers, according to some scale parameter R . This scale parameter delineates the scale-space boundary between the detail layer C_D and base layer C_B . To obtain the texture/object decomposition, we perform a texture simplification operation, which renders the object layer C_B , and subtract it from the original image to obtain the texture layer $C_D = C - C_B$ (figure 6.7). We will describe this texture simplification operation in more detail in the following section.

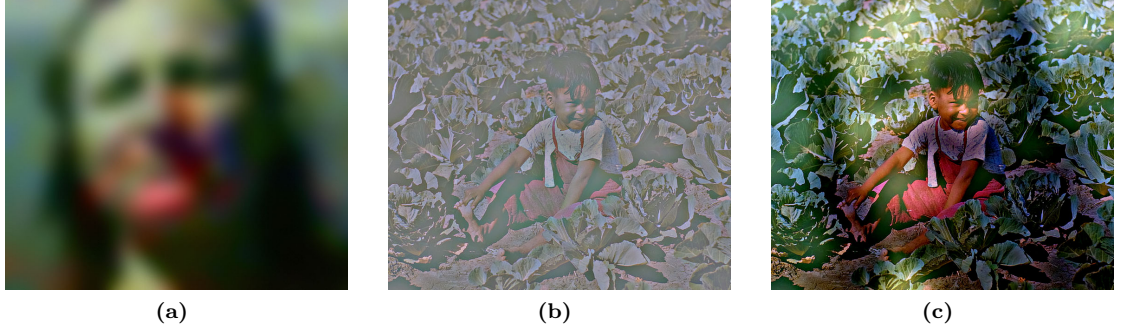


Figure 6.5: A hybrid image, using a method similar to the that presented in Hybrid Images [68]. **6.5a** Base layer, with colors adjusted to better suit the detail layer. **6.5b** Detail layer. **6.5c** Final hybrid image.

Once we have the object and texture layers C_B and C_D , we then cross-bilateral filter the target image T with the object layer C_B from the collage, again according to the scale parameter R . Recall that the bilateral filter operation, introduced to the image processing world by Tomasi and Manduchi [91], performs an image smoothing operation in which the smoothing kernel varies according to the content of the image at each position. Most often the kernel is, for each position, constructed in such a way that the weight of each pixel is somehow proportional to its distance in intensity from the center pixel. More precisely, the bilateral filter B of image I may be described as such:

$$\begin{aligned}
 B(I, x, y) = & \frac{1}{N(I, x, y)} \sum_{\Delta x, \Delta y} f_a\left((x, y), (x + \Delta x, y + \Delta y)\right) \\
 & \cdot f_b\left(I(x, y), I(x + \Delta x, y + \Delta y)\right) \\
 & \cdot I(x + \Delta x, y + \Delta y)
 \end{aligned} \tag{6.3}$$

where $f_a(., .)$ and $f_b(., .)$ are positive-valued distance functions. The normalization factor $N(I, x, y)$ is defined as such:

$$\begin{aligned}
 N(I, x, y) = & \sum_{\Delta x, \Delta y} f_a\left((x, y), (x + \Delta x, y + \Delta y)\right) \\
 & \cdot f_b\left(I(x, y), I(x + \Delta x, y + \Delta y)\right).
 \end{aligned} \tag{6.4}$$

The cross-bilateral filter, introduced by Petschnigg et al. [74] and Eisemann and Durand [31], is a bilateral filter operation in which the filter kernel for each pixel is computed from an image different than the image being filtered; essentially, the bilateral influence of the filter comes from another image:

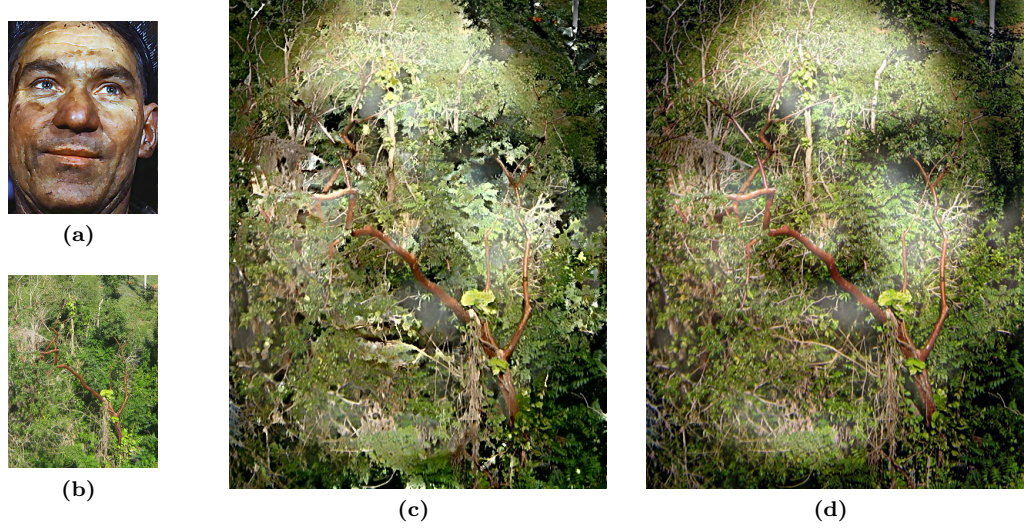


Figure 6.6: A comparison of two double image methods. Our method, shown in figure 6.6c, shows sharper edges and shared features between the secondary and primary images. Haloing is largely absent from the primary image, due to the edge-preserving nature of the decomposition. However, it is due to this that some distracting distortions to the primary image have appeared that are not present in the original method, such as inverted gradients in the child’s arm and face. Figure 6.6d shows a double image composed by a method similar to the Hybrid Image method of Oliva et al. [68]

$$\begin{aligned}
B_{\text{cross}}(I, J, x, y) = & \frac{1}{N(I, x, y)} \sum_{\Delta x, \Delta y} f_a \left((x, y), (x + \Delta x, y + \Delta y) \right) \\
& \cdot f_b \left(J(x, y), J(x + \Delta x, y + \Delta y) \right) \\
& \cdot I(x + \Delta x, y + \Delta y)
\end{aligned} \tag{6.5}$$

where image J is used to determine the filter kernel at each pixel.

Petschnigg et al. and Eisemann and Durand used the cross-bilateral filter for image denoising when combining flash and no-flash photograph pairs using a base/detail approach. Similarly, in our system this cross-bilateral operation acts as a sort of low-pass filter, blurring the image I but preventing blurring over edges in image J . We may use the result as our new base layer P_B .

$$P_B = B_{\text{cross}}(T, C_B, x, y) \tag{6.6}$$

This base image P_B from the target image is blurred in a way that respects the edges of the objects in the collage. When combined with the texture high-pass layer C_D from the collage, we see in figure 6.8 that the result is an image that more closely resembles the original source image, but does not exhibit any obvious signs of direct combination between the target and the collage. Note that we have not directly used the base layer of the target image T_B in computing P_B ; as always, in every step of the collage generation process, we do not wish to directly use the target



Figure 6.7: **6.7a** The collage image. **6.7b** The abstracted collage image after texture simplification.

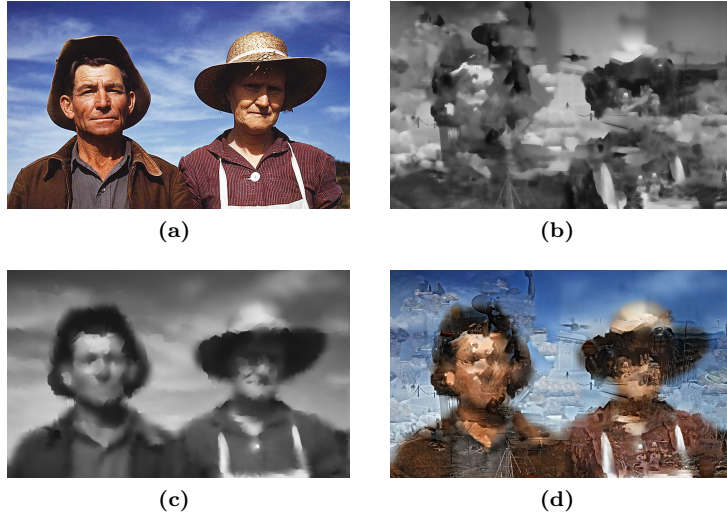


Figure 6.8: **6.8a** The original target image T . **6.8b** The texture-simplified collage image. **6.8c** The cross-bilateral result. **6.8d** The cross-bilateral result combined with the texture high-pass from the collage.

image T in any obvious way, but only to approximate it as an aggregation of objects and other image features from our library. We then have, for the final color corrected image P :

$$P = P_B + C_D. \quad (6.7)$$

The process described thus far is summarized in figure 6.9.

6.3.1 Detail Enhancement

While the image P is certainly an improvement on the directly computed collage C , we may be able to do even better. In the main part of this section, we have concentrated on ensuring that the object layer of the collage matches as closely as possible the target image T . We have left the texture layer of the collage, C_D untouched, but we may also modify it to better represent the details of the target image. We have accomplished this by augmenting C_D with an additional detail layer,

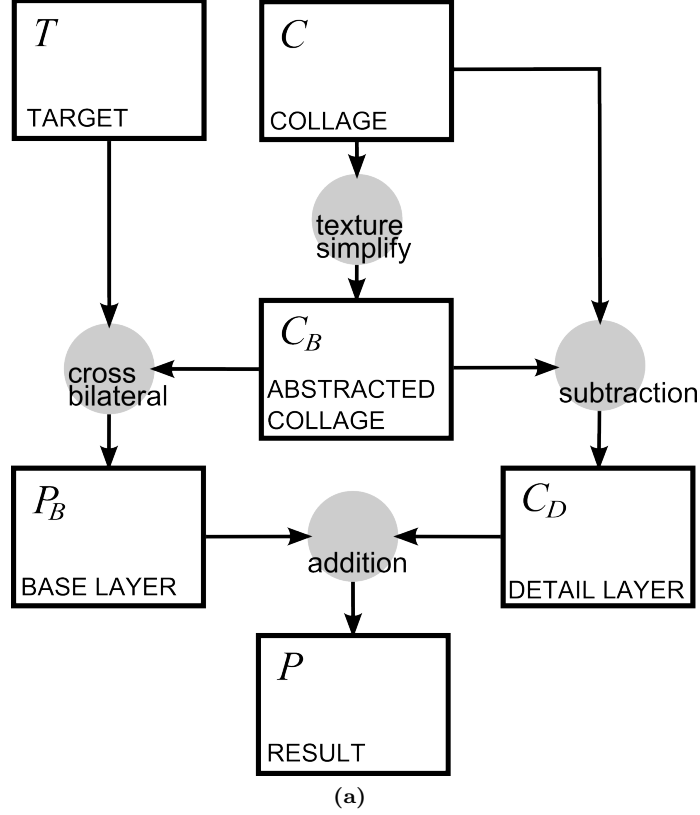


Figure 6.9: A diagram of the color correction post processing.

P_0 , which we compute in the following way:

$$P_0 = \tau(C) \left[B_{\text{cross}}(T, C) - G_r(T) \right], \quad (6.8)$$

where G_r is a Gaussian smoothing operation for some small radius r determined by the scale of the details we wish to represent, and where it is understood that $B_{\text{cross}}(T, C) = B_{\text{cross}}(T, C, x, y)$. What we are doing here is filtering the target image bilaterally across the collage, as we did to produce P_B , but for a smaller blur radius; then, we are performing a Gaussian high-pass filter to extract the resulting details, so that we can add them to the final result. The effect of this operation is a sort of distortion, as a scene is distorted when viewed through textured glass. In this way, we bend the details of T to fit the details of C , and the addition of these bent details back into the collage acts to accentuate the details of C in such a way that they better suggest the details of T . In a final tying up of loose ends, we multiply P_0 by the texture contrast $\tau(C)$ of the collage, so that we do not introduce any new detail to the final collage where before there was none. The effects of this detail enhancement are illustrated in figure 6.10. The final result may then be expressed as follows:

$$P_{\text{final}} = P_B + C_D + P_0. \quad (6.9)$$

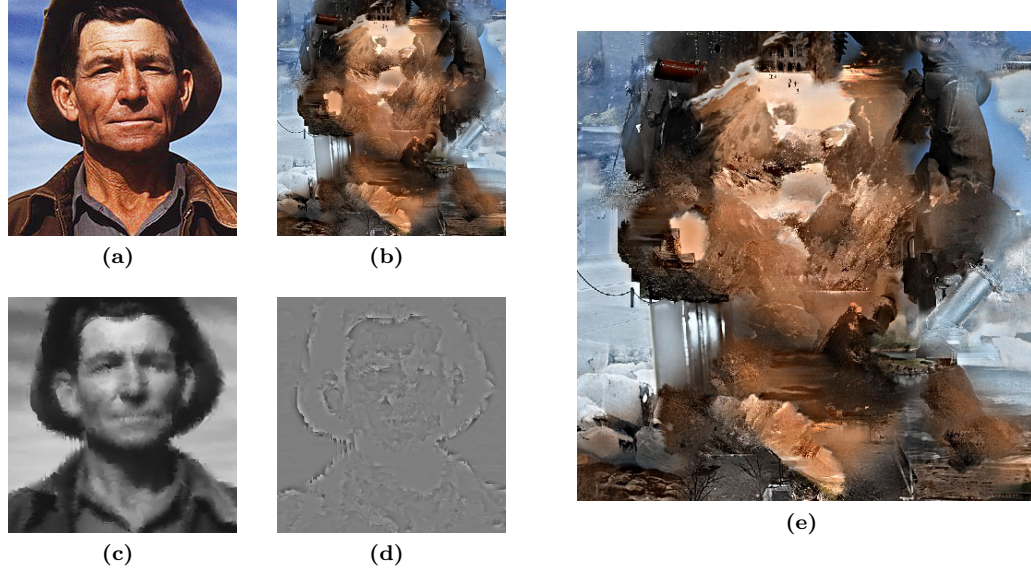


Figure 6.10: **6.10a** The target image T . **6.10b** The color corrected collage. **6.10c** The Gaussian high-pass of T . **6.10d** The high-pass after bilateral filtering with C . **6.10e** The final result $P = P_B + C_D + P_0$. Notice the details of the eyes and mouth have naturally been brought out of the collage images.

Parameter Tuning Given the large number of parameters involved in the creation of a collage such as tile scale, image scale with respect to tiles, and the nature of the image library itself, it is difficult to determine a set of parameters for postprocessing that works for every collage. We have found that using a uniform scaling parameter for all tasks (within a factor of 2), and choosing the scale parameter to match the scale of the collage tiles, provides a good base (as we see in figure 6.10), but parameter tweaking is invariably required to achieve the best look. Certainly, there is an art to this process, and other image processing operations may be used to enhance the result further. For example, in chapter 7, figure 7.17 we have mixed the postprocessed result with the original collage to accentuate shadows in the library images. Furthermore, different scale parameters may be used to produce each of the base and detail layers.

6.3.2 Texture Simplification

Previous Investigations in Edge-preserving Smoothing Perona and Malik [73] provided an early solution to the problem of edge-preserving smoothing. Their method of anisotropic diffusion uses a series of iterated regularization operations [81] to simulate diffusion, while preventing diffusion across image object boundaries in order to preserve edges. Since the introduction of anisotropic diffusion, it has been pointed out [91] that its iterative nature poses problems of stability and efficiency. As is noted in Farbman et al. [33], the bilateral filter, first introduced by Tomasi and Manduchi [91], is commonly used as an image abstraction or texture simplification filter. However,

Bae et al. [3] point out that the bilateral filter will not sufficiently handle high-contrast, high-frequency features, which we wish to simplify along with low-contrast, high-frequency features. Of course, the trick is that we also wish to preserve high-contrast edges, and in order to do this we must distinguish them from high-contrast textures.

Choudhury and Tumblin [15] improve the performance of the bilateral filter in high gradient regions by tilting (or rather, skewing) the filter kernel in the intensity dimension according to the gradient vector of a bilaterally filtered version of the image. Fattal et al. [34], for shape and detail enhancement, use an iterated bilateral filter, reducing the depth of the filter at each iteration, and increasing the spatial radius with each iteration, in order to achieve a smoother, deeper filter. However, the case of high-contrast textures is not addressed in either of these methods.

The Weighted-Least-Squares (WLS) method of Farberman et al. [33] is extremely effective in abstracting images to an object layer and texture layer, including the abstraction of high-frequency, high-contrast texture data. However, the resulting base layer seems to suffer from an over-flattening of important gradients which serve to convey object shading and lighting. Comaniciu and Meer [17] introduce the Mean-Shift filter, a filtering operation for image smoothing and abstraction based on the mean-shift segmentation operation. However, this method of filtering suffers from a similar drawback to WLS.

The methods found in the realm of image abstraction, described in chapter 2, may be applied to the problem of texture simplification. Particularly, the method of Orzan et al. [70] seems to exhibit good texture simplification properties without specifically setting out to separate textures from objects. This method is aimed at abstracting out edges that do not persist through a deep range of scales, and many textures are composed of such edges, although not all; textures of the highest amplitude may not be abstracted by this operation.

Any of these methods, including the basic bilateral filter and perhaps even the antiquated anisotropic diffusion, may be used to obtain the base and detail layers required by the methods of the main part of this section, with some degree of success. However, our method is an effective alternative and, to the best of our knowledge, is the only method that specifically aims to remove textures, of most varieties, from objects.

A Novel Approach As we have stated, the bilateral filter is a convolution operation in which the convolution kernel varies as a function of position in the image and the image content within the neighborhood of the position. However, the standard bilateral filter is insufficient for abstracting textures with high contrast, because the high contrast features are indistinguishable from image edges by the standard bilateral filter. Our texture simplification operation consists of a modified bilateral filter operation, in which the depth, or the influence of the intensity component, is modulated by a scalar map $\tau(C)$ representing texture contrast. This method is reminiscent of the work

of Su et al. in de-emphasizing regions of high texture activity [88]. The effect of this is that blurring will occur not only over low contrast texture regions, but also over higher contrast texture regions, effectively removing textures while preserving object edges. See Tomasi and Manduchi's original paper [91] for a complete discussion of the bilateral filter. The texture simplification filter we use may be more precisely expressed as follows:

$$\begin{aligned}
B_{\text{texture}}(I, x, y) = & \frac{1}{M(I, x, y)} \sum_{\Delta x, \Delta y} f_a\left((x, y), (x + \Delta x, y + \Delta y)\right) \\
& \cdot \left(1 - \tau(I(x, y))\right) \cdot f_b\left(I(x, y), I(x + \Delta x, y + \Delta y)\right) \\
& \cdot I(x + \Delta x, y + \Delta y)
\end{aligned} \tag{6.10}$$

where

$$\begin{aligned}
M(I, x, y) = & \sum_{\Delta x, \Delta y} f_a\left((x, y), (x + \Delta x, y + \Delta y)\right) \\
& \cdot \left(1 - \tau(I(x, y))\right) \cdot f_b\left(I(x, y), I(x + \Delta x, y + \Delta y)\right)
\end{aligned} \tag{6.11}$$

Alternately, we have achieved good results using a standard cross-bilateral filter, with the bilateral component composed of a blend between the target image and a Gaussian blurred version of the target image, with the blending amount controlled by the texture contrast map. This allows the smoothing algorithm to be easily implemented with standard filters. More precisely:

$$\begin{aligned}
B_{\text{texture}}(I, x, y) = & \frac{1}{N(I, x, y)} \sum_{\Delta x, \Delta y} f_a\left((x, y), (x + \Delta x, y + \Delta y)\right) \\
& \cdot f_b\left(I'(x, y), I'(x + \Delta x, y + \Delta y)\right) \\
& \cdot I(x + \Delta x, y + \Delta y)
\end{aligned} \tag{6.12}$$

where

$$I' = \left(1 - \tau(C(x, y))\right)I + \left(\tau(C(x, y))\right)G_{\sigma_r}(I) \tag{6.13}$$

The method for obtaining a texture contrast map is explained fully in chapter 3. In brief, we exploit the fact that the median filter, while exhibiting some level of preservation of step edges, completely flattens even high contrast textures. By taking the absolute differences between an image and its median-filtered companion, and iterating this operation, we may obtain a novel and useful measure of texture contrast. Figure 6.11 shows a comparison of some image abstraction filters with our own filter.

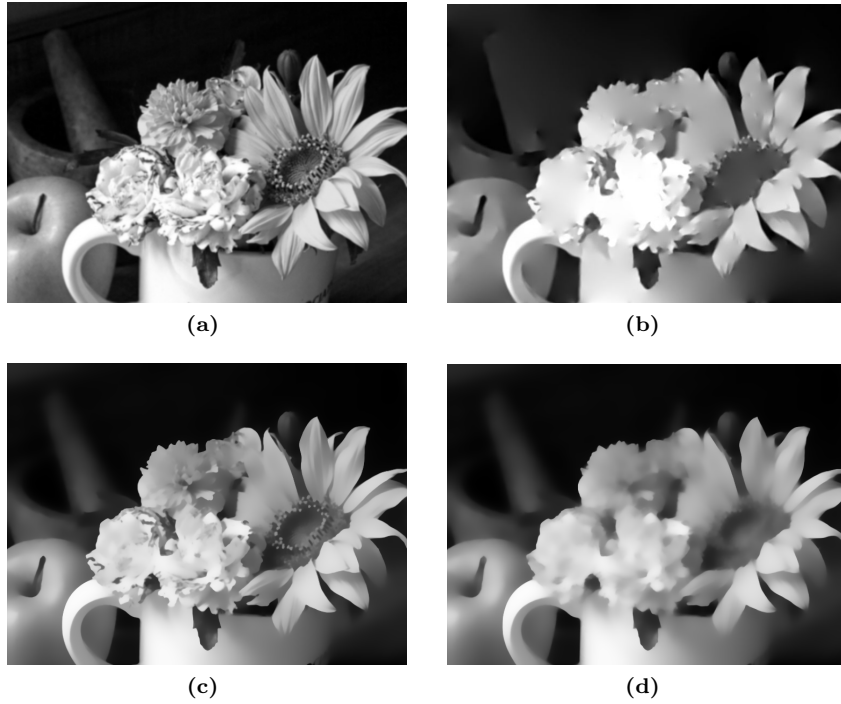


Figure 6.11: **6.11a** Original Image. **6.11b** Image abstraction via the method of Orzan et al. [70]. **6.11c** Image after ordinary bilateral filtering (2 iterations). **6.11d** Image after texture simplification via our method (2 iterations).

CHAPTER 7

RESULTS AND EVALUATION

In this chapter we will discuss the products of the collage system described heretofore and evaluate each component of the processes involved. First we will review and summarize the algorithmic elements introduced thus far. Then, as part of this discussion and evaluation, we will present various examples of each extrema in the space of parameters involved in the collage creation process, as well as some typical and best-case examples. We will then discuss failure cases and open issues with the system as it stands.

7.1 Overview of the Algorithm

In this section we bring together all of the system elements we have introduced throughout the previous chapters, and review the entire algorithm.

Preprocessing We begin by extracting from the target image edge and salience maps for future use, as well as a texture contrast map. Each of these maps must also be downsampled for multiscale processing.

- Edge maps are produced for various scales. The purpose of these edge maps is, for each scale, to weight regions of the target image near strong edges as less significant in the calculation of match quality. In this way, we may improve collage discernibility and prevent over-refinement. For a given scale r , a gradient magnitude map is taken as an initial starting point for the edge map. The corner map of this gradient magnitude map is taken, to remove unwanted texture edges. The edge map $F(I)$ of image I should have range $[0, 1]$ where lower values represent stronger edges; therefore, we subtract the entire result from 1. That is,

$$F''_r(I) = 1 - \text{Corner}_r(|\nabla G_r(I)|), \quad (7.1)$$

where G_r represents a normalized Gaussian smoothing operation of radius r .

The higher scales should also include the edges from the lower scales, since we wish to respect the content that has been already laid out. So, for each scale after the first and coarsest, we multiply each edge map produced so far.

$$F'_S(I) = \prod_s^S F''_s(I) \quad (7.2)$$

Also, we prefer that pixels in empty regions of the image, far from edges and other salient features, have less weight. Therefore, we extend the edge map F' by some radius R representing the falloff distance of the weight map, by taking a normalized Gaussian blur filter of the edge map and then multiplying it with the original edge map F' . We also enforce some minimum weight c for empty regions by adding it as a constant.

$$F_S(I) = G_R(F'_S(I)) \cdot F'_S(I)(1 - c) + c \quad (7.3)$$

- A salience map S is produced for the purpose of signifying salient features of the target image that do not correspond to strong step edges. We use this map to ensure that the tiles chosen for refinement lie over salient data and not in empty regions of the image. The salience map is composed as a simple corner map of the target image, for some radius r (chosen according to the size of the target image and the scale the details the user wishes to highlight). The salience map increases in value with increasing salience:

$$S(I) = \text{Corner}_r(I). \quad (7.4)$$

- A texture contrast map τ_r , for some radius r , is computed for the target image and each library image; for the purposes of speed, we calculate these maps at a low resolution. Since we calculate this map at a low resolution, we also calculate only one iteration of the map:

$$\tau_r(I) = M_r(|I - M_r(I)|), \quad (7.5)$$

where M_r represents the median filter of radius r .

- For the target image and each library image, the luminance, chrominance, and texture contrast channels are all downsampled for each level of resolution used in the matching process (this is two levels of resolution for our tests).
- Histograms are calculated for each channel of each library image, for the library reduction stage of the matching process. We have used 8 bin histograms for the purpose of speed.

Matching Once we have done the necessary preprocessing of the target image and library images, we then create a partition of the target image for each iteration of refinement we wish to perform, and then begin matching each tile in the coarsest partition with some image from the library. After creating an entire coarse-scaled collage, we repeat the process for a partition with smaller tiles, and

only for those tiles that lie above a part of the collage that was poorly matched in the previous stage. Every tile, except those that have not been chosen for refinement, should be matched with an image from the library, with an associated positioning coordinate.

- A partition of the target image is produced by calculating a centroidal Voronoi diagram, using the edge map $F(I)$ as the mass function. A Voronoi diagram is a partition of a space, according to some given set of centroids, such that every point within a tile is closer to the centroid associated with that tile than to any other centroid. A *centroidal* Voronoi diagram is one in which the centroid for each tile lies at the center of mass for some function. We calculate this partition by first calculating a Voronoi diagram, moving each of the centroids to the center of mass for each tile, and then recalculating the Voronoi diagram. This operation is typically performed iteratively until convergence is achieved (this is called Lloyd's Algorithm). We use seven iterations, which we have found, through experimentation, produces regions with the desired characteristics in a short period of time.
- The region of the target image contained within each tile is matched against the library to find a representative collage image.
 - First, we begin by reducing the size of the library by keeping only those images in the library that have many pixels near the mean value of the target image pixels within the current tile. We determine which images these are by analyzing their histograms, which have been computed in preprocessing.
 - Then, we do a pixel-wise comparison of each image in this reduced set with the tile content. We also compare every translation of a given image against the content of the tile. We keep the best matches from this process, further reducing the library, and repeat the process again at a higher resolution. This is repeated again until we have reached the desired maximum resolution, and then the best image is chosen as the image with the lowest pixel wise error.

An image I , for a given translation and channel $n = \{l, c, t\}$ for luminance, chrominance or texture contrast, is ranked according to how well the following quantity is minimized:

$$E_n = \sum_{(x,y) \in A} F(x,y)M(x,y)|I_n(x,y) - T_n(x,y)|, \quad (7.6)$$

where $A = I \cap T$ indicates the rectangular region of intersection between image I and image T and $M \in [0, 1]$ is a fraction-valued function representing the tile mask. The quantity F represents an edge map for strong step edges.

For an image I and a given translation, the final ranking is determined by the following quantity:

$$E = w_l E_l + w_c E_c + w_t E_t, \quad (7.7)$$

where $\vec{w} = (w_l, w_c, w_t)$ are parameters chosen by the user to influence the character of the final result; specifically, the three parameters (w_l, w_c, w_t) influence the weight of luminance, chrominance, and texture contrast, respectively, in the matching calculation.

Compositing At the end of the matching process, every tile (except for those tiles lying above sufficiently accurate portions of the collage) will have an image and a set of translation coordinates associated with it. Then, it is time to draw the collage as it currently stands. At this point, we recalculate the Voronoi partition (using the positions of the centroids initially calculated) using path cost rather than the Euclidean norm to measure the distance from each pixel to each centroid. The path cost is calculated through a weight map that is derived from a combination of edge maps from each collage image to be composited.

The contribution to the weight map of each image is computed as a sum of gradient magnitude edge maps over several scales:

$$e(I) = \sum_{i=0}^M |\nabla G_{2^i} * I|. \quad (7.8)$$

The contributions of each image are combined into one final edge map for the entire collage. The pixels from each image edge map are weighted proportionally by their distances from the edge of the tile boundary. Where there is overlap between tile images, the maximum of the edge values is taken.

$$e_{\text{final}}(T, x, y) = \text{Max}(e_i(I_i(x, y)) \cdot v(x, y)_i), \quad (7.9)$$

for all edge maps e_i and where $v(x, y) \in [0, 1]$ is proportional to the distance from a boundary in the original Voronoi Tessellation, with greater weight closer to the boundary. we calculate v as follows:

$$v(x, y) = G_R(\text{TileBoundary}(x, y)), \quad (7.10)$$

where R is a scalar on the order of the distance across which tile images are blended, and G_R is the normalized Gaussian blur operation.

Refinement Following the composition of the collage, at each intermediate stage, an error map must be computed so that decisions can be made about which tiles should be calculated in the next refinement iteration. Again, as in the matching stage, the error map is multiplied with the edge map $F(I)$ and also with the salience map $S(I)$. Then, the error density of each tile in the next partition is calculated, and the tiles with the highest error density are flagged for recalculation. The number of tiles that are flagged for recalculation at each stage are limited to a certain proportion.

Postprocessing At this point, we have a finished photocollage. Now we wish to improve the accuracy of the collage by applying color correction. We produce the final, color corrected collage by composing a base layer, derived from the target image and collage, with a detail layer derived from the collage and with detail accents added in from the target image.

- We begin composing the base layer P_B by first abstracting the collage C (removing textures and other fine details) to obtain an image that roughly represents the image objects of the collage. Then we cross bilateral filter the target image T with this abstracted collage image:

$$P_B = B_{\text{cross}}(T, C_B), \quad (7.11)$$

where $B_{\text{cross}}(.,.)$ represents the cross-bilateral filter, and $C_B = \text{TextureSimplify}(C)$ represents the base layer of the collage.

- We create the detail layer $P_D = C_D$ by abstracting the collage, as in the above step, and then subtracting it from the original collage to separate the textures and details from the rest of the image. The base and detail layer are additively combined:

$$P = P_B + C_D, \quad (7.12)$$

where $C_D = C - \text{TextureSimplify}(C)$.

- Finally, we attempt to tweak the details of the collage so that they are able to also represent the details of the target image. First, we duplicate the process for creating the base layer, but on a much finer scale. Then, we take a Gaussian high pass filter of this result P_0 and add it to the rest of the collage.

$$P_0 = \tau(C) \left[B_{\text{cross}}(T, C) - G_r(T) \right], \quad (7.13)$$

$$P_{\text{final}} = P_B + C_D + P_0. \quad (7.14)$$

P_{final} represents the final color corrected collage.

7.2 Discussion of Results

The production of a photocollage depends on many parameters and is accomplished through various different systems, depending on different image understanding and rendering frameworks. Similarly, the scale for measuring success is also quite varied. We have attempted to isolate many sensible

components of the image collaging process and analyze them independently. We also have made some general observations of phenomena that tend to arise in the production of photocollages. Following is a qualitative analysis of the known problems in photocollage and image mosaic systems, the efficacy of our proposed solutions, and phenomena we have observed through the course of our investigation.

7.2.1 Selection of Target Image

As with most NPR systems, there are some images that are more amenable to the process than others. We have had the most success with images that have the following properties:

- A steep difference between the texture contrast of adjacent image regions. We use fairly low-level methods for treating texture, and adjacent regions with similar texture properties tend to get muddled together in the final collage;
- High contrast, large scale objects, which are few in number and perceptually disjoint. Images that contain large aggregations of small, low contrast objects tend to produce muddled collages. An image consisting of an aggregation of objects is what we are trying to *produce*, not what we wish to start from;
- Natural, organic or photographic image content. Line art and vectorized images can produce interesting collages, but as with some other NPR processes, hard edges and constant colors are not well-preserved.

See figure 7.1 for an example of a target image for which our algorithm produces an unsatisfactory collage.



Figure 7.1: A poorly produced collage and the corresponding target image. **7.1a** Original target image. The image contains multiple objects with low contrast. **7.1b** The photocollage. Notice that not even color correction has been able to bring the collage up to an acceptable standard.

7.2.2 Accuracy

While our method is able to produce collages of reasonable accuracy in a small amount of processing time, we have observed that the collages are not as accurate as they are discernible. We have attempted to address this through postprocessing without compromising discernibility. The advantage of this approach is that large gains in accuracy can be achieved without increasing the size of the image library or decreasing the coarseness of the tiling. In fact, one can achieve high accuracy even with a library of only a few images and a very coarse tessellation (figure 7.2).

When viewed from perspective of double images, as discussed in chapter 1, it comes down to a comparison of two distinct philosophies: an exhaustive search through a large database for a better solution, or tweaking and manipulating an existing solution to make it better. For one-to-one double images, a combination is used; a human artist, employing some creative process, is able to choose an arrangement of objects that is able to convey the secondary, target image ideally while also rotating, scaling and otherwise distorting these objects as necessary.

Another observation is that some damage to perceptual accuracy seems to occur as a result of the multiscale nature of the collaging process itself. While seams between tiles have been hidden to some degree, it is often apparent, to a human eye, where one tile ends and another begins simply through context; a night scene composited next to a day scene shows clearly the difference between two tiles. The eye is thus guided through a visual interpretation of the collage by the size and placement of tiles in relation to each other, and the essence of the target image is not always respected; that is, the scale of tiles used may not be congruent with the scale of objects in the target image, and thus the target image may be difficult to discern. The scale of objects in the target image is usually pulled back into the collage through post processing; even so, we hypothesize that the preprocessed collage is often less accurate to the human eye than it is in terms of pure pixel-wise error.

7.2.3 Discernibility

We have chosen to forego the use of content based region of interest detection when determining the valid space of translations for each library image, in hopes of improving accuracy. While we have achieved some success with this method, as evidenced by the images we have presented, there have been some drawbacks. Particularly, it is common for images consisting largely of textures to be chosen, and purely textured regions from these images to be used in the final collage. While this occurrence may not be noticeable in traditional image mosaics, such as Orchard and Kaplan’s cut-out image mosaics [69], due to the small size of the matching primitives and clear distinction between tiles, it sometimes becomes a problem when larger tiles are used. It may not even be entirely accurate to classify this as a discernibility problem, since these textures are certainly discernible;

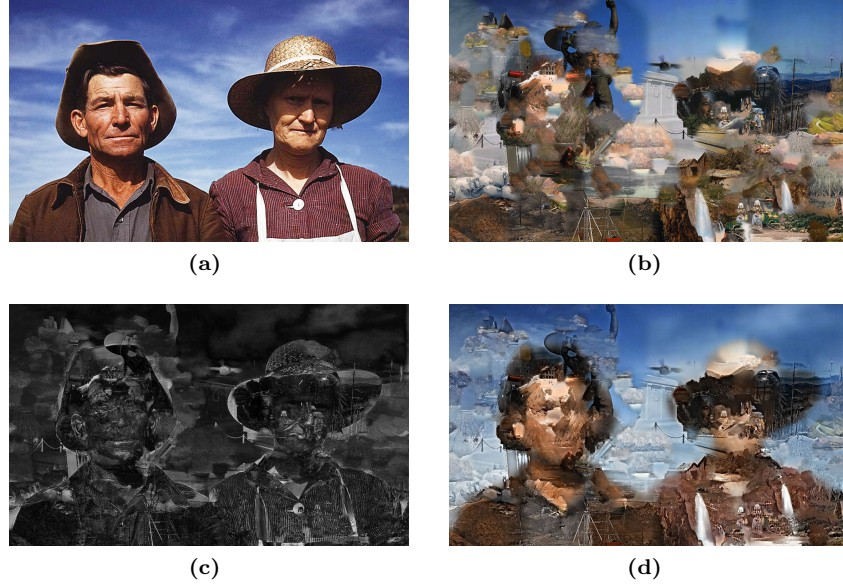


Figure 7.2: **7.2a** Target image. **7.2b** Photocollage. **7.2c** Absolute error between target image and collage. **7.2d** Photocollage after post processing.

they just aren't very interesting!

7.2.4 Compositing

When compared to the composition results of Autocollage [78], which uses graph cuts and Poisson image blending, our method holds up well, and shows superior performance in some cases as shown in figure 7.4. Autocollage shows no significant advantage over our method, even though it is significantly more complex. In figure 7.5 we see that each method similarly preserves image content while smoothly changing between images.

Some minor problems still persist. For instance, haloing occurs around some hard edges, as a thin strip of the background of one tile persists along the tile's edge. Also, the shapes of some image objects appear negatively in the composition; that is, the shape of a tile boundary is defined by an object that has been removed from the image through the compositing process, resulting in a mysteriously shaped tile, as we see in figure 7.3.

7.2.5 Image Library

For conventional, regular lattice image mosaics, Tran et al. [92] have shown that mosaic accuracy improves as a function of image library size. This is certainly true of photocollages as well; however, one of the benefits of our method is that collage accuracy is not as tightly correlated to library size. Iterative refinement, accuracy enhancement via post processing, and the expansion of the match space due to image translation all allow for a lower boundary on accuracy that is higher than in

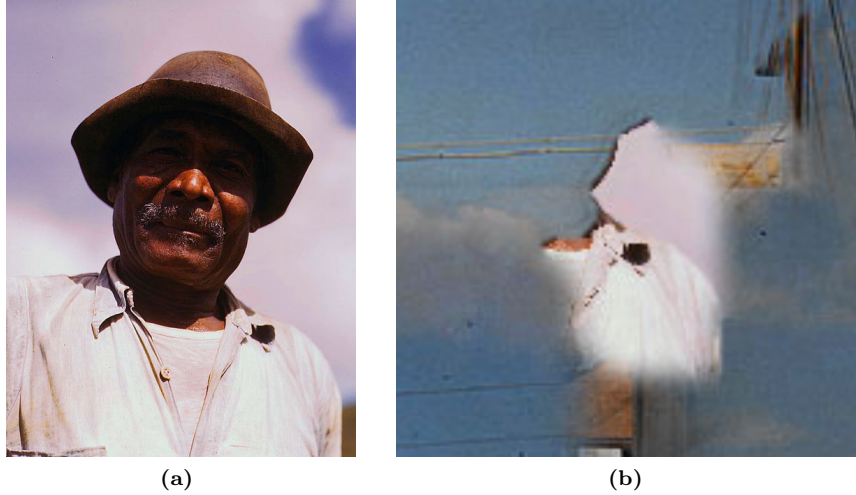


Figure 7.3: 7.3a Original library image. 7.3b Image in the composition. The face has not been included, but the shape remains as part of the tile boundary.



Figure 7.4: Two examples of successful compositions via path cost Voronoi tessellations.

the case of the standard image mosaic. Even with a very small library (or even a well-chosen single image!) an attractive and accurate collage may be produced. In section 7.2.6 we discuss the effective expansion of the image library resulting from scaling and translating the library images against the collage tiles.

Of course, there are also benefits to be had for using a very large image library. Figure 7.6 illustrates the effects of increasing the library size on the outcome of the collage process.

7.2.6 Tile Image Scale

We have discussed the effect of the scaling factor in chapter 5. Here we will examine the effects of some tile image scaling policies on the discernibility and accuracy of collages.

As we have discussed, the larger an image is with respect to the tile with which it is being



Figure 7.5: **7.5a** Our composition method. **7.5b** An image produced by Microsoft AutoCollage 2008 [79].

compared, the greater the potential accuracy of the final match. Orchard and Kaplan [69] noted that searching the space of translations against the tile region effectively increases the size of the image database being consulted for each match. For example, an image I , which is X times larger in each dimension than the current tile, will provide $4X^2$ possible matches, assuming that a translation of $X/2$, in any direction, is necessary to produce a sufficiently unique match (in practice, we use an even smaller step size for shifts). So, for a scaling factor $s = X$, the size of the image database is effectively multiplied by $4X^2$. Of course, one must look into the effect of s on the discernibility of resulting collages, since each of these “effective images” is more fragmentary and less complete than in the case of $s = 1.0$. Another thing that must be considered is the increased processing time for large s . Since we are increasing the effective size of our database, we can expect an increase in processing time due to a larger search space. However, the methods that we have employed to mitigate the high costs of searching this space, as described in chapter 4, are not effective for mitigating the costs associated with an enlarged search space due to a large scaling factor S . This is because these methods operate on the level of the image, and not on the sub-image level of image translations. So, these extra points in the search space have been smuggled in with each image.

As expected, we see in figure 7.7 that an increased scale factor leads to increased accuracy. Accordingly, processing time also increases with the scaling factor. When considering issues of discernibility, it is not clear that a larger scale factor is necessarily “better” than a lower one; the figures in section 7.5 show collages computed for scaling factors between 1.3 and 2.0, which have good accuracy and discernibility. For collages with three levels of refinement, the results indicate more of an advantage to using a larger scale factor, since the iterative refinement process is able to rectify some errors. In one case, an average decrease in absolute error of 7% was observed, along with an average increase in processing time of 56% when increasing the scaling factor from 1.5 to 3.0.



Figure 7.6: An illustration of the effect of library size on collage quality. **7.6a** Original Image. **7.6b** Photocollage with a library of 27, randomly selected images. **7.6c** Photocollage with a library of 79 randomly selected images. **7.6d** Photocollage with a library of 838 randomly selected images.

In chapter 4 we discussed the idea of adjusting the scale factor s for each detail level such that s is proportional to tile size. It seems that in this way we may shift the increased accuracy and processing time to areas of the image where it is most necessary. However, we found that when linearly interpolating between scaling factors of 1.5 and 3.0 as refinement increased, the processing time increased by an average of 38% while the quality increase was negligible. It is possible that for very large images, this strategy may provide a good compromise of efficiency, but for smaller images (between 500 and 1500 pixels in the longest dimension) it is not recommended.

7.2.7 Refinement and Scaling

We believe a large difference between the size of the largest tiles and the smallest is preferable, aesthetically. A wide variation of spatial scales contributes depth to the collage and engages the visual and interpretive senses of the viewer; a rich and dynamic space is established, and the depth of the image is expanded. In section 7.5 we show a sample of collages that have been computed with both one refinement pass and two refinement passes; there is an evident difference in the character of collages that are produced with a deeper range of scales, although we see that accuracy is very high for collages with a shallower range of scales. Practically speaking, the size of the tiles in

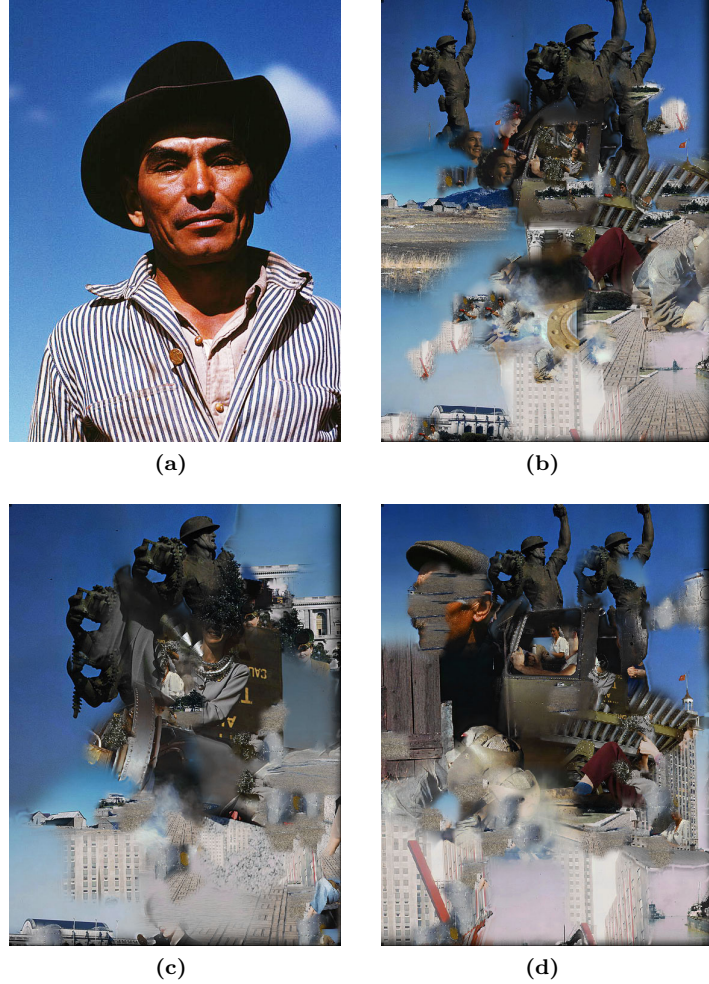


Figure 7.7: An illustration of the effect of image-size to tile-size ratio on *discernibility* and *accuracy*. **7.7a** Original Image. **7.7b** Collage with scaling factor $S = 1.5$ **7.7c** Collage with scaling factor $S = 3.0$ **7.7d** Collage with scale factor varying from $S = 1.5$ to $S = 3.0$ for the smallest tiles.

each refinement level, in proportion to the tiles in other refinement levels, determines some of the character of the accuracy/discernibility tradeoff in the final collage. We have found that a functional and visually effective policy for determining tile size is to reduce the average tile size, in the next refinement level, by a factor between 3 and 6. See figure 7.14 and figure 7.13 to compare the effects of different tile size proportions.

For target images whose size is on the order of the 1000×1000 , or roughly the resolution of a typical consumer display, three levels of refinement are usually appropriate (sometimes even two are sufficient). An additional level of refinement would require the use of tiles that would be indiscernible due to their small size. For larger images, of course, more levels of refinement may be appropriate.

7.2.8 Edge Weights in Matching and Refinement

As we have said, it is easy to achieve high accuracy at the expense of discernibility but difficult to achieve both high accuracy and discernibility at the same time. One method we have proposed for increasing discernibility at a modest cost in accuracy is, when evaluating the match cost of a particular tile image, to weight pixels around high contrast step edges as *less important*. The purpose of this is to allow minor deviations in the trajectory of step edges without undue penalization in the match cost calculation. This, in turn, will prevent (in the refinement stage) over-refinement of step edges, and (in the matching stage) tile matches that pave over the step with a solid tone.

We have shown some typical results of this strategy in figure 7.8. As you can see, the collage result without modified edge weights shows high accuracy to the original image, but also shows clusters of small, poorly discernible tiles around high contrast edges. The results with edge weighting show poorer accuracy, but much higher discernibility of tile images around edges. As you can see from the results in chapter 6, the application of base/detail postprocessing mitigates the decline in accuracy to some degree.

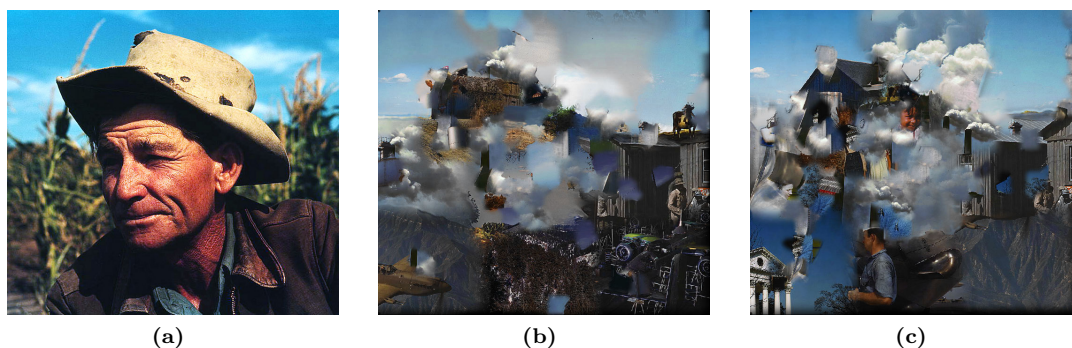


Figure 7.8: 7.8a Original target image. 7.8b Collage, without edge weights. 7.8c Collage, with edge weights. Note that the hard step edge on the brim of the hat has suffered less subdivision.

7.2.9 Corner and Saliency Maps in Refinement

When evaluating the error between the collage and the target image, it is common to discover that high errors reside in low activity, non salient regions of the image. This can lead to a large amount of computational resources being directed towards computing new matches for regions of the image in which new matches are not really needed. In these cases, the old matches, while not entirely accurate, are sufficient given the underlying content of the target image. As we said in chapter 4, we may avoid this by multiplying our error map by some map that measures the saliency of the target image, so that errors in low saliency regions are dampened. We discussed in chapter 4 the technique of applying a saliency map S that is more sensitive than the edge-weight map F , used

to increase discernibility, as described in chapter 4 section 4.3.2, so that it is not canceled out. We have found that in order to prevent the total cancellation of all errors outside the neighborhood of some salient image feature, it is useful to modify the salience map by raising the minimum value to some non-zero value.

In practice, the error map is dominated by the influence of the edge map F , even when the salience map is calculated at a finer scale. Furthermore, restricting the number of tiles to be refined in the next pass to some fixed proportion goes a long way to preventing unnecessary refinement (as opposed to using a fixed error density threshold). This makes sense, since an arbitrary number of tiles may have error density above a given threshold. Even so, the use of a salience map may be used to modestly improve the results in the case that a coarse scale edge map is not used.

7.2.10 Luminance and Chrominance

Since the human visual system is more sensitive to changes in luminance than changes in chrominance [90], it may be helpful to weight luminance data heavier in the calculation of matches. The expectation, in this case, is that the result should be more accurate in luminance, while the colors are less preserved. Orchard and Kaplan [69] take these properties into account when they suggest a single degree of freedom approach to color correction, in which only the luminance channel is shifted.

As you can see in figure 7.9, collages composed with low chrominance influence, specifically $w_l = 1.0, w_c = 0.2$, will not accurately represent the colors of the target image and may have a somewhat rainbow colored appearance; this is mitigated somewhat through postprocessing. It is not sufficient to ignore chrominance altogether in the matching process, since large adjustments to chrominance in postprocessing become obvious and detrimental to the quality of the collage. Chrominance adjustments that change the color of objects so that they are obviously wrong according to semantic classification by user (for example, the blue cows observed in figure 6.2), and adjustments that split an object into several, apparently arbitrary and disjoint color regions, are undesirable. Therefore, some attention to chrominance is necessary during the matching process, although overall, higher perceived accuracy may be achieved by emphasizing luminance, as shown in figure 7.9.

7.2.11 Texture Contrast

In chapter 4, we discussed the problem of smooth areas in the target image being matching with non-smooth images in the collage. We have attempted to address this problem by adding a texture contrast term to the error energy calculation in the matching process. In this, it appears that we have achieved a modest level of control over the smoothness of the collage result. In figure 7.10 we show the results of a collage computed with high texture contrast influence, and a collage computed without texture contrast influence. As you can see, the texture contrast term has shifted

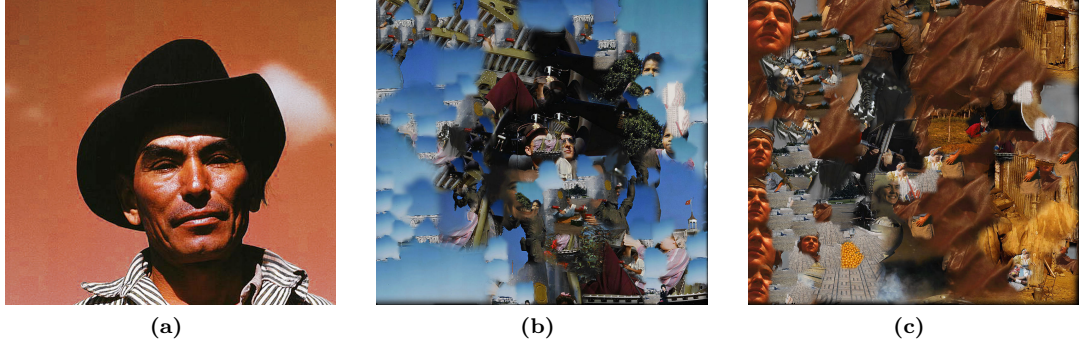


Figure 7.9: The effect of changes to luminance/chrominance balance. **7.9a** Target image. **7.9b** Low chrominance influence ($w_l = 1.0, w_c = 0.2$). **7.9c** Balanced chrominance and Luminance ($w_l = 1.0, w_c = 1.0$).

the priority from tone matching to matching the smoothness of the target image.

However, there are some problems with the current approach that prevent it from being as useful as color and luminance in image matching; since the primary result of matching with texture contrast is to enforce smoothness, discernibility often suffers, since low-activity and uninteresting image regions are used. Also, we do not consider the scale or orientation of the textures; therefore, while textures of similar contrast may be matched, the discrepancy in scale may render this match invalid. This is complicated by the fact that each image is rescaled in proportion to each tile, so the texture scale is not fixed for a given image. In light of this, we recommend using texture contrast as a matching term only when it is imperative to maintain smoothness in the collage result. Even in these cases, the results appear to be largely dependent on the scale parameters used and the content of the image library.

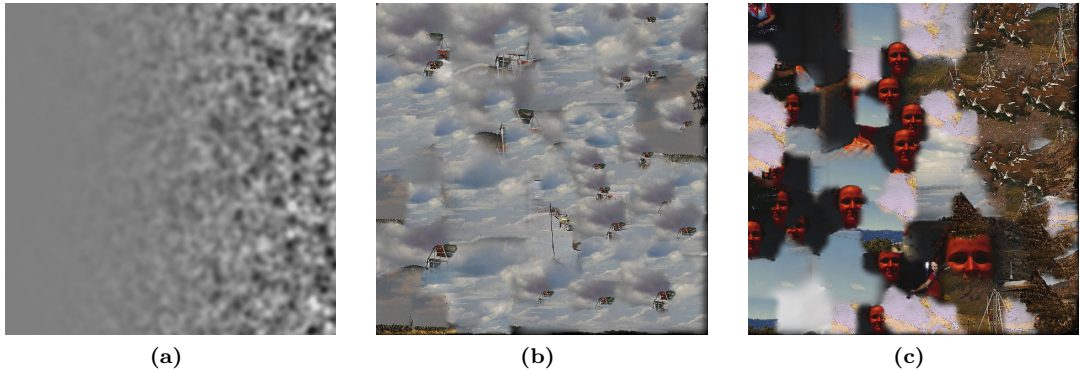


Figure 7.10: The effect of changes to the balance of texture contrast in the matching calculation. **7.10a** Target image. **7.10b** No texture influence ($w_l = 1.0, w_c = 1.0, w_t = 0.0$). **7.10c** No luminance or color influence ($w_l = 0.0, w_c = 0.0, w_t = 1.0$). Smooth, relatively low contrast images are used for the left-hand side of the image.

7.2.12 Library Reduction

Before matching the library images directly with the target image, we first reduce the size of the image library to a subset of likely good matches. As described in chapter 4, this is done by comparing the mean value of the target image inside the current tile with the histogram of each library image.

To measure the effectiveness of this strategy we compared the collage error and processing time of collages computed with and without library reduction (the library being reduced by 90%). Using a scaling factor of $s = 1.3$, we found that using library reduction increased the average collage error to 104% of the error without library reduction, and reduced the processing time to 89%. With a scaling factor of $s = 2.0$, the error was increased to 109% of the error without library reduction, and the processing time was reduced to 77% of the time to compute a collage without library reduction. The scaling factors were chosen to represent typical “high discernibility” and “high accuracy” parameters, respectively. These results indicate that library reduction has less effect for images that are scaled close to the size of the tiles, but is useful for reducing the processing time when the scaling factor is high, even though accuracy is decreased. For comparison, library reduction by random image selection caused the error to increase to 138% and 142% for each scaling factor, with near identical processing time.

7.2.13 Scale Hierarchy

After the matching process is completed for a given resolution, some proportion of the best matches must be chosen to go onto matching at the next higher resolution. We have chosen, for the sake of efficiency and timely processing, to process two scale levels after the initial library reduction pass. For the first pixel-wise matching pass, we use an image resolution of $1/16$ the original, and for the second pass we use a resolution of $1/4$ the original. As for the proportion of images used from each pass, we have found that the processing time may increase dramatically with higher proportions, depending on the resolution. This makes sense, since the time complexity of a pixel-wise image comparison across the space of translations is quadratic. For the resolutions we have chosen, the second pixel-wise comparison stage will have $4 \times 4 = 16$ times more comparisons; then, if we wish for the second stage to take the same amount of time as the first, we must compare $1/16$ of the images (this assumes a constant scaling factor). For a small library, this may be too few. We have opted to use a uniform proportion of 6% for the final, most precise matching pass, which is small enough to keep processing time down, but large enough to provide a sufficient number of potential matches for even a small library.

As for the first pixel-wise matching pass after the initial library reduction stage, we have found that there is little difference in collage quality (either in perceived accuracy or absolute error)

between collages computed with a reduction to 33% of the original library and 50%, despite a significant increase in processing time (22% in a typical case). This is not surprising, but indicates that the ordering of images from the first matching pass is a good approximation to the ordering that will come from the next pass. Of course, the ordering will not be identical; if it were, one pass would be sufficient.

7.2.14 Postprocessing

Overall, the use of the postprocessing algorithm described in chapter 6 is able to dramatically increase the accuracy of a collage. Even a collage which initially exhibits very poor accuracy (due to a small library or other factors) may be repaired in this way. Perceptual accuracy is improved through detail enhancement, as we see in figure 7.11.



Figure 7.11: **7.11a** Detail of photocollage, before post processing. **7.11b** Detail of photocollage, after post processing. Note that discernibility of library images remains high. **7.11c** Detail of target image.

Since much of the low spatial frequency data of the collage has been replaced with data from the target image, strange results sometimes may be observed, particularly, reversal of gradients. This has been observed by other investigators in the area of edge-preserving decompositions [33]. However, this does not occur often or conspicuously enough to warrant significant investigation within this project, since the matching process generally produces a collage with low frequency data sufficiently similar to that of the target image (see figure 7.12). Furthermore, the original low frequency data from the collage may be blended with the base layer (generated by the cross-bilateral operation) in a sufficient proportion as to avoid obvious gradient reversals and other visually distracting anomalies. The best proportion in which to blend the original and new base layers varies widely with the image being processed; the new base layer should be roughly twice as prominent as the old base layer, in order to ensure that the postprocessing is effective.

When determining the scale of the cross bilateral filter used to produce the base layer, there

must be some consideration of the nature of the target image. Generally, we recommend that a scale be used that is on the same order as the scale of the most prominent tiles, within a factor of two. If the width of the blur is too small, it is possible that the resulting gradients across collage image objects may be too steep in some cases; that is, image objects in the collage which fall across step edges in the target image may be spuriously bisected in the postprocessed result. We wish, instead, to augment the collage result with gentle, wide gradients that maintain the spatial coherence of image objects.

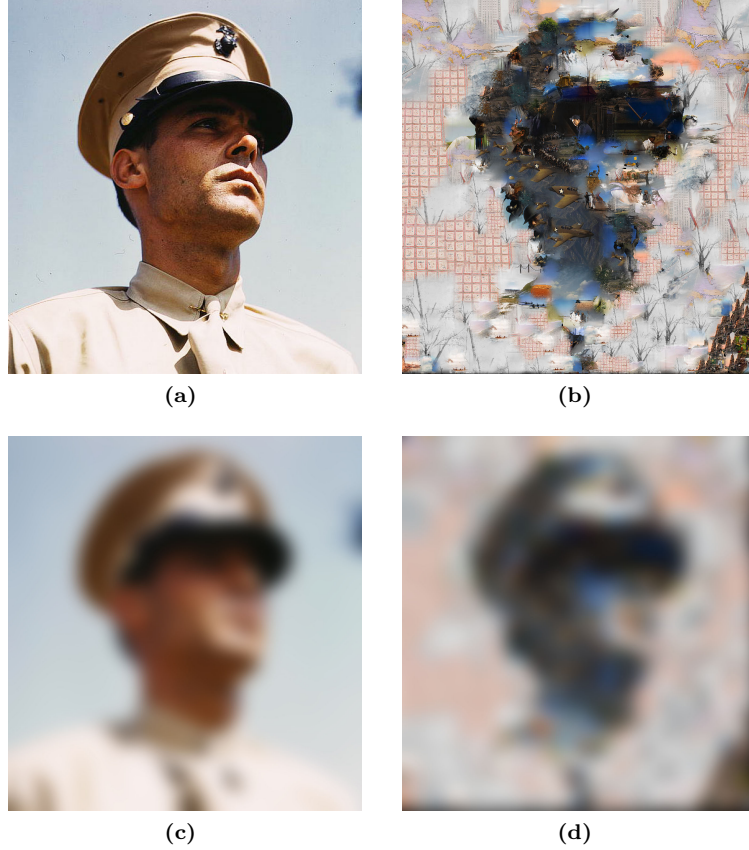


Figure 7.12: **7.12a** Target image. **7.12b** Collage, before postprocessing. **7.12c** Lowpass of target image. **7.12d** Lowpass of collage.

7.3 Efficiency and Performance

Throughout the entire design process of this system, we have kept speed and efficiency as a primary goal. When designing a system that is meant to process arbitrarily large sets of data, this is a necessity; a collaging system is not useful if it can only be used, within a reasonable amount of time, on a very small database. We have achieved a system that is both straight-forward in implementation, effective, and relatively fast. The hierarchical matching scheme is naturally

applicable to processing image data, which is known to be well represented by multiscale and pyramidal structures [13]. We have also designed our system in such a way as to minimize the amount of library queries that must be carried out; by working to make the most from large tiles through localized refinement and clever post-processing, we are able to cut down the number of tiles that must be matched.

For a target image of size 1400×945 pixels, with a database of 79 images, each of size $655 \times 500 \pm 50$ pixels, the time for determining all matches was 17.0 minutes. The image preprocessing step took 0.84 minutes. Calculating the time for two more databases (of size 53 and 27 images) we found that the average rate of matching was 13 seconds per file. For a database of 422 images, as Orchard and Kaplan use for their tests, this indicates the processing time would be 1.5 hours. This test was conducted with two additional refinement passes after the first and a uniform scaling factor of 2.0 across refinement levels. The proportion of tiles calculated was 33% and 25% for each of the two refinement passes respectively. All the rest of the parameters were set as recommended in the previous part of this chapter. The computer used to perform these tests was a Dell XPS 420, with the Intel Core 2 Duo 3.00Ghz processors and 3.00Gb of RAM. Our implementation has been programmed in Java 6.0.

Orchard and Kaplan report, for their Cut-Out Image Mosaic method [69], a time of 4.7 hours, using a database of 422 images, each of size 60×80 on an Apple Macintosh 2.50Ghz PowerPC G5 workstation with 8.00Gb of RAM. They do not report the size of the target image, so it is difficult to compare this performance with ours. In any case, the method of Orchard and Kaplan is different enough in its fundamental philosophy that it is difficult to make a direct comparison. We have designed our method so that in the worst case fewer match calculations are necessary.

The calculation of texture contrast, edge maps, corner maps, and salience maps all have been done through methods which may be implemented as a combination of fast image processing methods, such as the $O(1)$ bilateral filter of Fatih Porikli [75] (which extends to other filters, such as the median filter), and fast approximation methods such as those proposed by Paris and Durand [71]. The same is true for the post processing operations, which also may be implemented through known fast methods. For the tests reported above, using an unoptimized implementation, the average time for the calculation of everything other than the image/image comparisons was 12.9 minutes (this time is independent of library size).

7.4 Failures and Limitations

While we have made significant progress towards an effective photocollage system, there remain some aspects of the process that may be improved upon. In this section, we discuss certain failure cases of our proposed system, limitations that extend across all cases, and suggestions for possible

remedies. First, we will discuss the main problems with the system as it stands.

Unpredictability Due to the randomized nature of tile selection, several collages for a given target image and library often vary widely in quality. In chapter 4, we discussed options for generating the collage tiles; many of these do not depend on randomization, so it may be fruitful to investigate these further. Along a similar track to the centroidal Voronoi tessellations, it is possible that using edge segments (from some binary edge detection mechanism) instead of single points may yield favorable results.

Repetition of Images We have chosen to allow the repetition of images in the composition of the collage. We felt that the multi-scale nature of the tiling, as well as the placement of the tiles across salient image regions, would naturally discourage the occurrence of multiple adjacent tiles being rendered with the same image. Orchard and Kaplan [69] identified the problem of a flat-shaded region being paved with multiple copies of the same image; but if the sky is represented by only a few tiles, this is less likely to occur, and if these tiles contain salient image structures from other parts of the image, it is even less likely. Also, the fast matching system allows for the use of large databases, which also may discourage repetition.

In practice, repetition among adjacent tiles does occur frequently. The initial matching step—that is, the reduction of the database by histogram comparisons—often will lead to several adjacent tiles having very similar sub-libraries to draw from, and thus repetition occurs. This somewhat mitigates the advantage of having a large database. Perhaps more sophisticated methods of image clustering, such as those proposed by Oliva and Torralba [67], may be used to rectify this problem.

Refinement Efficiency We have observed that there is often a high proportion of overlap in the image regions that are refined at each level. That is, any pixel in the image is most likely to either be recalculated several times, or not at all. This seems to point to a problem with algorithm efficiency; it is reasonable to expect that a region may need to be calculated more than once, but it seems like a waste to recalculate very similar stretches of the image over and over. It is possible that these regions simply are more difficult to match, and thus must be reprocessed several times. More investigation into this problem is required.

Selection of Parameters Even though, through extensive experimentation, we have gathered a set of operation parameters that are appropriate for general use, it is usually possible to improve the visual quality for any given target image through further exploration of the parameter space. The problem is that there are many parameters on which the quality of the end result relies. We have outlined the most important of these parameters in this section, but there are others. For instance, the post-processing step alone contains a scale parameter for the the production of the

detail layer, two scale parameters for the production of the base layer (one for the width of the cross-bilateral filter, and one for the image abstraction of the cross-image), scale parameters for the detail layer, parallel parameters for each color channel and various blending parameters for mixing all these components together. As a general policy, we have used scale parameters that are uniform across all tasks (within a given refinement level). However, invariably, tweaking is required to get the best results for each target image. A more extensive investigation is required to determine a comprehensive and strict system for determining parameters, if one even exists.

7.5 Selected Results



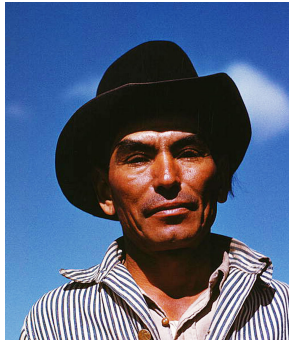
(a)

Figure 7.13: A collage emphasizing discernibility, made from 685 images. Individual collage images are circled.



(a)

Figure 7.14: A collage emphasizing accuracy, made from 685 images. Individual collage images are identified.



(a)



(b)

Figure 7.15: A collage emphasizing discernibility. The size of the library is 837 images, and two refinement passes are used.



(a)



(b)

Figure 7.16: A collage emphasizing accuracy. The size of the library is 837 images, and one refinement pass is used.

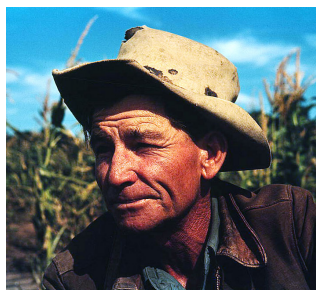


(a)

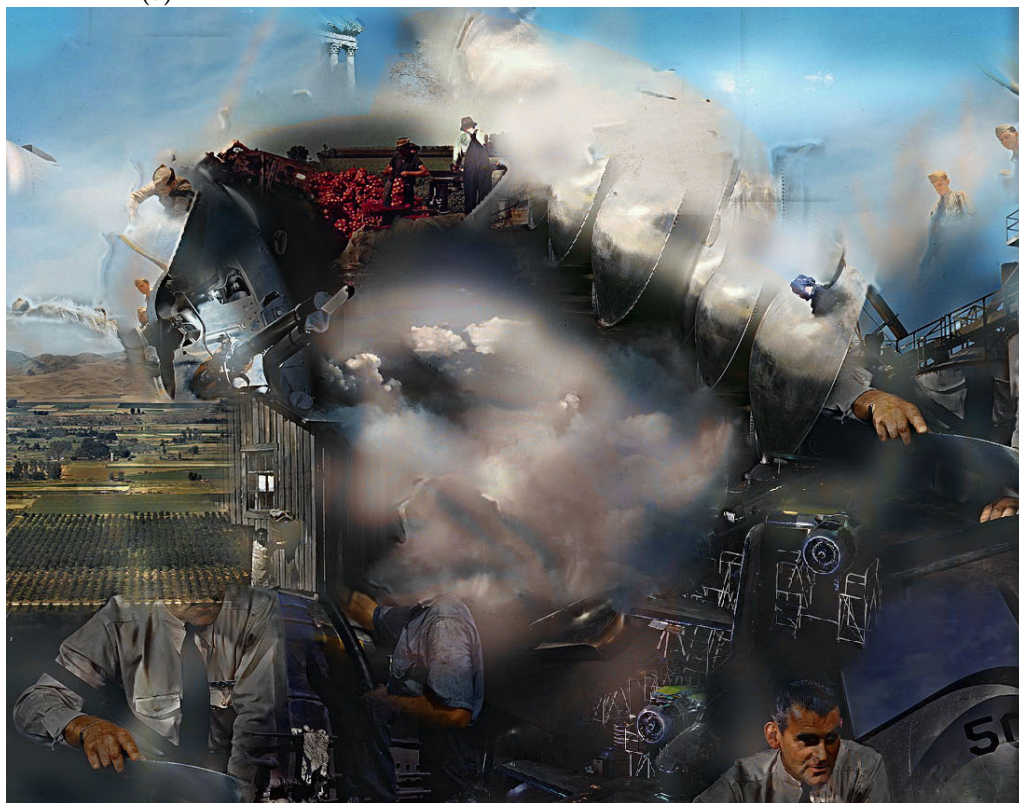


(b)

Figure 7.17: A collage made from 30 images of the author, with one refinement pass. Additional post processing was done to emphasize edges in the library images.



(a)



(b)

Figure 7.18: A collage emphasizing discernibility. 837 images are used, and one refinement pass.



Figure 7.19: A collage emphasizing discernibility, made from 685 images. Saturation has been enhanced to improve discernibility.



(a)



(b)



(c)

Figure 7.20: A collage made from 685 images, with two refinement passes. **7.20b** The result of Orchard and Kaplan [69] is shown for comparison.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

Contributions Orchard and Kaplan produce image mosaics using tiles of irregular shape; however, the shapes of their tiles have nothing to do with the content of the images composited therein, or even the target image. Rother et al. render collages with consideration of image content, without attempting to match any target image. We combine these two approaches to create collages that contain fascinating interactions between adjacent collage images, and between the collage images and the target image. The individual images of the collage flow from one to the next, without any distracting tile seams to break up the image. As in image mosaics, the photocollage shows interesting correspondences between the collage images and the target image, but more so; since we have made the use of large tiles a priority and facilitated the use of large databases, there is less reliance on arrays of small tiles each containing poorly matched images.

All in all, multiscale image processing has been the backbone of the matching algorithm. Not only have we been able to maintain high discernibility by using large tiles, and using smaller tiles only when needed, but we have been able to accelerate the search process through the image library by matching low resolution images first and then increasing the resolution as needed. The whole algorithm brings together various ideas inspired by signature matching, image registration and content based image retrieval.

We have tackled the problem of achieving high accuracy and discernibility simultaneously within a single collage. By localizing the criteria for measuring the fulfillment of these requirements, we have allowed for collages with increased discernibility while maintaining an acceptable level of accuracy. Over all, we have found that it is easiest to keep discernibility high during the image matching and compositing process, while attending to the accuracy requirements in postprocessing. This seems to be parallel to what is seen in the photographic mosaic techniques that currently exist. The matching process and color correction processes work together to create double images that are comparable to those produced by artists, although they certainly have their own character and charm.

We have experimented with the use of image edge and salience maps to influence the outcome

of the matching process in favor of accuracy or discernibility, depending on the underlying features of the target image. By allowing looser matches in the neighborhood of strong step edges, we have been able to create collages with high discernibility in areas of the image where heavy subdivision is often used to maintain accuracy. Similarly, by also allowing looser matches in empty, flat regions of the image, and limiting the number of tiles that may be refined, we've reduced the incidence of over-refinement in non salient image regions.

We have also experimented with different weights for each of the terms in the calculation of the matching score for determining the best image for a given tile. We have shown that favoring luminance over chrominance is a way to achieve higher perceptual accuracy while still maintaining some correspondence between the colors in the target image and the collage. Also, we have tried using a texture contrast as a matching feature, and found it somewhat useful in representing smooth regions of the target image with smooth collage images. However, for general images, the use of the texture contrast term generally does not improve the results significantly.

Color post-processing has been used in various image mosaic systems to pull up accuracy after the main matching process has been completed. Our system also relies on color correction, but draws from hybrid image techniques to obtain a good balance of accuracy and discernibility in the final result. The base/detail approach and the use of edge preserving decompositions has allowed us to render both primary and secondary images viewable at any distance by tying together image features across scales. Additionally, by tweaking the detail layer of the collage, we have been able to effectively transmit detail information through from the target image to the collage without harming the discernibility of the collage images.

Throughout the process of developing the techniques heretofore described, we have also come across some interesting image processing operations that have allowed us to produce salience maps, texture contrast maps, and abstracted images in unique ways. By recognizing the particular responses of texture images under the application of median filters, and drawing on previous approaches to texture contrast measurement, we have developed a class of filters that may certainly be built upon in further investigations of texture analysis, image abstraction and edge preserving decompositions.

— -

Context Many forms of double image have been thought up by artists, and many more certainly will be thought up in the future. The introduction of algorithmic methods for automatically composing double images may already be contributing to the introduction of new forms, even though computational methods in this areas are relatively new. In any case, computational methods have certainly made a once laborious and time consuming task—the task of creating image mosaics and collages—something potentially achievable by any user, be the user an artist, graphic designer or

amateur tinkerer. The creativity of artists can never be predicted, but only assisted. Artists will find new ways to combine images and make unique statements by creating new visual vocabularies. We may leave it up to them to develop new ways of combining images, and then try to assist them by automating these processes. However, we may also be able to assist in the creation of new forms by allowing artists greater freedom to experiment. We hope that artists will use this photocollage tool in ways that we have not anticipated. We hope also that it will be used to inspire new types of double images, motivating further research and development in this area.

The photocollage system presented here allows users to create complex collages from any library of images, to match any target image. Given the large number of parameters involved in the whole process, there is the potential for a wide range of looks that may be achieved with some experimentation. The user may trade between accuracy and discernibility, sharp or soft tile boundaries, and large or small collage images. For a release version of the software, it is unclear what the best set of parameters would be to make available to the user, but there is certainly some flexibility in this area. The system is relatively fast, and thus accessible to anyone with a standard desktop computer.

8.2 Summary of Future Work

In this section, we offer some brief suggestions for developing further the system that we have introduced in this thesis.

Image Features We have shown that a scale-based matching process is effective for querying a large database for image regions with certain feature characteristics. These results may be extended in a great number of ways; an obvious direction for future research is the exploration of expanded feature sets.

We have proposed a few basic methods for using the corner map and texture contrast map described in chapter 3 for image abstraction and texture simplification. However, we believe there exist more sophisticated approaches that may be discovered through further investigation. Additionally, it seems likely that there exist more sophisticated approaches for calculating corner maps and texture contrast maps using more detailed and complete statistical structures than simple grey level histograms.

Double Images More investigation into the methods that artists use to create double images is warranted. In this thesis, we have assumed a method of distorting and modifying the structures of the secondary image in order to convey it with a more discernible primary image. However, it appears that many artists, including such artists as Ocatavio Ocampo and Salvador Dali, actually prefer to distort the primary image in order to more accurately convey the secondary image. In our

method, we modify the primary images by cutting them from their natural contexts and compositing them into the collage, and then applying pixel-wise color transformations in order to improve accuracy. More work may be done in applying spatial distortions to the matching primitives in the primary image to improve accuracy, with modest discernibility costs. The authors of Jigsaw Image Mosaic [51] and Puzzle Image Mosaic [9] have already explored these ideas.

Collage Tiles We have identified several methods for producing tilings, each of which have advantages and drawbacks. We believe further investigation is warranted into these and other methods of producing tilings, as well as other methods of modifying each tiling for iterative refinement. Particularly, there are likely many methods of producing edge-straddling segmentations that may be explored. We suggest the use of Voronoi regions with edges as generators; we touched on this idea briefly in chapter 5. Bilateral Voronoi Regions may also be useful, as used by Inoue and Urahama [49].

We have devised a system for using path cost Voronoi diagrams to find the optimal composition of a set of images, given their scaling and translation transforms, and the region centroids. However, there may be many variations of this technique which could produce various different types of images, depending on the application. We encourage the adaption of these techniques to other NPR realms. Particularly, path cost Voronoi tessellations may be useful for seam repair in patch based texture synthesis.

Discernibility Measurement Throughout this book, we have used a subjective method for judging discernibility. It may be possible to judge discernibility in a quantitative way, to some degree, by analyzing the content of image tiles.

Texture Contrast In order to improve the response of our texture contrast filter to a wider range of texture scales within a single image, it may be useful to use automatic scale selection techniques as employed by Carson et al. [14] and others.

REFERENCES

- [1] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. In *SIGGRAPH '04: ACM Transactions on Graphics (TOG) 2004 Papers*, pages 294–302, New York, NY, USA, 2004. ACM.
- [2] Michael Ashikhmin. Synthesizing natural textures. In *I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 217–226, New York, NY, USA, 2001. ACM.
- [3] Soonmin Bae, Sylvain Paris, and Frédo Durand. Two-scale tone management for photographic look. *ACM Trans. Graph.*, 25(3):637–645, 2006.
- [4] Nick Bantock. Nickbantock.com. Website, July 2009. <http://www.nickbantock.com/index.html>; Last accessed: 02/11/2010.
- [5] P. Bao and Lei Zhang. Scale correlation-based edge detection. In *Video/Image Processing and Multimedia Communications 4th EURASIP-IEEE Region 8 International Symposium on VIPromCom*, pages 345–350, 2002.
- [6] Sebastiano Battiato, Gianpiero Blasi, Giovanni Gallo, Giuseppe Claudio Guarnera, and Giovanni Puglisi. A novel artificial mosaic generation technique driven by local gradient analysis. In *ICCS '08: Proceedings of the 8th international conference on Computational Science, Part II*, pages 76–85, Berlin, Heidelberg, 2008. Springer-Verlag.
- [7] Sebastiano Battiato, Gianpiero Di Blasi, Giovanni Maria Farinella, and Giovanni Gallo. A Novel Technique for Opus Vermiculatum Mosaic Rendering. In Joaquim Jorge and Vaclav Skala, editors, *Proceedings of the 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2006, February, 2006, Plzen, Czech Republic)*, pages 133–140, Plzen, 2006. University of West Bohemia, UNION Agency.
- [8] Sebastiano Battiato, Gianpiero di Blasi, Giovanni Maria Farinella, and Giovanni Gallo. A survey of digital mosaic techniques. In Giovanni Gallo, Sebastiano Battiato, and Filippo Stanco, editors, *Eurographics Italian Chapter Conference*, pages 129–135. Eurographics, 2006.
- [9] Gianpiero Di Blasi, Giovanni Gallo, and Petralia Maria. Puzzle image mosaic. In *In proceedings of IASTED/VIIP2005*, 2005.
- [10] Gianpiero Di Blasi, Giovanni Gallo, and Maria Pia Petralia. Smart ideas for photomosaic rendering. In G. Gallo, S. Battiato, and F. Stanco, editors, *Proceedings of Eurographics Italian Chapter Conference*, 2006.
- [11] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, Nov 2001.
- [12] Philip Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover Publications, New York, 1966.
- [13] Peter J. Burt and Edward H. Adelson. The Laplacian pyramid as a compact image code. *Readings in computer vision: issues, problems, principles, and paradigms*, pages 671–679, 1987.

- [14] Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1026–1038, 1999.
- [15] Prasun Choudhury and Jack Tumblin. The trilateral filter for high contrast images and meshes. In *SIGGRAPH '05: ACM ACM Transactions on Graphics (TOG) 2005 Courses*, page 5, New York, NY, USA, 2005. ACM.
- [16] Christopher M. Christoudias. Synergism in low level vision. In *ICPR '02: Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02) Volume 4*, page 40150, Washington, DC, USA, 2002. IEEE Computer Society.
- [17] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
- [18] R. W. Connors, M. M. Trivedi, and C. A. Harlow. Segmentation of a high-resolution urban scene using texture operators. *Computer Vision, Graphics, and Image Processing*, 25(3):273–310, 1984.
- [19] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–721–II–728 vol.2, June 2003.
- [20] Ignacio Gomez de Liano. *Dali*. Rizzoli International Publications Inc., 712 Fifth Avenue/New York 10019, 1982.
- [21] Andre Pieyre de Mandiargues. *Arcimboldo the marvelous*. Harry N. Abrams, Inc., New York, 1977.
- [22] Doug DeCarlo and Anthony Santella. Stylization and abstraction of photographs. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 769–776, New York, NY, USA, 2002. ACM.
- [23] N. Diakopoulos, I. Essa, and R. Jain. Content based image synthesis. In *Conference on Image and Video Retrieval (CIVR)*, pages 299–307, Dublin, Ireland, July 2004.
- [24] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, December 1959.
- [25] Yoshinori Dobashi, Toshiyuki Haga, Henry Johan, and Tomoyuki Nishita. A method for creating mosaic images using Voronoi diagrams. In *Proceedings of Eurographics 2002*, September 2002.
- [26] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. *ACM Trans. Graph.*, 22(3):303–312, 2003.
- [27] Qiang Du, Maria Emelianenko, and Lili Ju. Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations. *SIAM J. Numer. Anal.*, 44(1):102–119, 2006.
- [28] Qiang Du, Faber V., and Gunzburger M. Centroidal Voronoi tessellations : Applications and algorithms. *SIAM J. Numer. Anal.*, 41(4):637–676, 1999.
- [29] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*, pages 341–346, August 2001.
- [30] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision*, pages 1033–1038, Corfu, Greece, September 1999.
- [31] Elmar Eisemann and Frédo Durand. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.*, 23(3):673–678, 2004.

- [32] Gershon Elber and George Wolberg. Rendering traditional mosaics. *The Visual Computer*, 19:67–78, 2003.
- [33] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans. Graph.*, 27(3):1–10, 2008.
- [34] Raanan Fattal, Maneesh Agrawala, and Szymon Rusinkiewicz. Multiscale shape and detail enhancement from multi-light image collections. In *SIGGRAPH '07: ACM ACM Transactions on Graphics (TOG) 2007 papers*, page 51, New York, NY, USA, 2007. ACM.
- [35] Geisa Martins Faustino and Luiz Henrique de Figueiredo. Simple adaptive mosaic effects. *Computer Graphics and Image Processing, Brazilian Symposium on*, 0:315–322, 2005.
- [36] Adam Finkelstein and Marisa Range. Image mosaics. In Roger D. Hersch, Jacques Andr, and Heather Brown, editors, *Electronic Publishing, Artistic Imaging and Digital Typography, Proceedings of the EP98 and RIDT98 Conferences, St Malo: March 30 - April 3, 1998, Lecture Notes in Computer Science Series, number 1375*, pages 11–22. Springer-Verlag, 1998.
- [37] Ran Gal, Olga Sorkine, Tiberiu Popa, Alla Sheffer, and Daniel Cohen-Or. 3D collage: expressive non-realistic modeling. In *NPAR '07: Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 7–14, New York, NY, USA, 2007. ACM.
- [38] Bogdan Georgescu and Chris Christoudias. Edge detection and image segmentation system (EDISON)v1.1. Software, 2002.
- [39] Bruce Gooch and Amy Ashurst Gooch. *Non-Photorealistic Rendering*. AK Peters, Ltd., 2001.
- [40] Derek Gores. Derek gores: Original artwork. Website, November 2009. <http://www.derekgores.com/collage.php>; Last accessed: 02/11/2010.
- [41] Nouchine Hadjikhani, Kestutis Kveraga, Paulami Naik, and Seppo Ahlfors. Early (m170) activation of face-specific cortex by face-like objects. *Neuroreport*, 20(4):403–407, March 2009.
- [42] Paul Haeberli. Paint by numbers: Abstract image representation. In *Proceedings of SIGGRAPH 1990*, August 1990.
- [43] Alejo Hausner. Simulating decorative mosaics. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 573–580, New York, NY, USA, 2001. ACM.
- [44] James Hays and Alexei A. Efros. Scene completion using millions of photographs. In *SIGGRAPH '07: ACM ACM Transactions on Graphics (TOG) 2007 papers*, page 4, New York, NY, USA, 2007. ACM.
- [45] I. Herman and D. Duke. Minimal graphics. *Computer Graphics and Applications, IEEE*, 21(6):18–21, Nov/Dec 2001.
- [46] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 453–460, New York, NY, USA, 1998. ACM.
- [47] Aaron Hertzmann. A survey of stroke-based rendering. *Computer Graphics and Applications, IEEE*, 23(4):70–81, July-Aug. 2003.
- [48] Helen Hutton. *The Technique of Collage*, chapter Part one: Techniques, pages 11–13. B.T. Batsford Ltd, Watson-Guption Publications, 1968.

- [49] Kohei Inoue and Kiichi Urahama. Generating stained glass-like images by bilateral Voronoi tessellation(<special section >computer graphics). *The journal of the Institute of Image Information and Television Engineers*, 61(10):1467–1471, 20071001.
- [50] Charles E. Jacobs, Adam Finkelstein, and David H. Salesin. Fast multiresolution image querying. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 277–286, New York, NY, USA, 1995. ACM.
- [51] Junhwan Kim and Fabio Pellacini. Jigsaw image mosaics. *ACM Trans. Graph.*, 21(3):657–664, 2002.
- [52] Junhwan Kim and Fabio Pellacini. Jigsaw image mosaics. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 657–664, New York, NY, USA, 2002. ACM.
- [53] Ken Knowlton. Knowlton mosaics: Computer assisted portrait art by computer graphics pioneer Ken knowlton. Website, October 2009. <http://www.knowltonmosaics.com/>; Last accessed: 02/16/2010.
- [54] I. Kokkinos, G. Evangelopoulos, and P. Maragos. Texture analysis and segmentation using modulation features, generative models, and weighted curve evolution. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(1):142–157, Jan. 2009.
- [55] Nikos Komodakis. Image completion using global optimization. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 442–452, Washington, DC, USA, 2006. IEEE Computer Society.
- [56] R. Kumar, H.S. Sawhney, J.C. Asmuth, A. Pope, and S. Hsu. Registration of video to geo-referenced imagery. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 2, pages 1393–1400 vol.2, August 1998.
- [57] Peter Litwinowicz. Processing images and video for an impressionist effect. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 407–414, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [58] Yu Liu, Olga Veksler, and Olivier Juan. Simulating classic mosaics with graph cuts. *Lecture Notes in Computer Science*, 4679/2007:55–70, 2007.
- [59] M.A. Lozano and F. Escolano. *Progress in Pattern Recognition, Speech and Image Analysis*, chapter Two New Scale-Adapted Texture Descriptors for Image Segmentation, pages 137–144. Springer Berlin / Heidelberg, 2003.
- [60] Trân-Quân Luong, Ankush Seth, Allison Klein, and Jason Lawrence. Isoluminant color picking for non-photorealistic rendering. In *GI '05: Proceedings of Graphics Interface 2005*, pages 233–240, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.
- [61] Lee Markosian, Michael A. Kowalski, Daniel Goldstein, Samuel J. Trychin, John F. Hughes, and Lubomir D. Bourdev. Real-time nonphotorealistic rendering. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 415–420, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [62] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., New York, NY, USA, 1982.
- [63] P. Meer and B. Georgescu. Edge detection with embedded confidence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(12):1351–1365, 2001.

- [64] Grant Morrison(w) and Dave McKean(a). *Arkham Asylum: A Serious House on Serious Earth*, chapter 2, page 36. DC Comics, College Station, Texas, 1989.
- [65] David Mould. Image-guided fracture. In *GI '05: Proceedings of Graphics Interface 2005*, pages 219–226, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.
- [66] The Library of Congress. The Library of Congress’ photostream. Website, 2011. http://www.flickr.com/photos/library_of_congress/; Last accessed: 02/18/2010.
- [67] Aude Oliva and Antonio Torralba. Building the gist of a scene: the role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006.
- [68] Aude Oliva, Antonio Torralba, and Philippe G. Schyns. Hybrid images. *ACM Trans. Graph.*, 25(3):527–532, 2006.
- [69] Jeff Orchard and Craig S. Kaplan. Cut-out image mosaics. In *NPAR '08: Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, pages 79–87, New York, NY, USA, 2008. ACM.
- [70] Alexandrina Orzan, Adrien Bousseau, Pascal Barla, and Joëlle Thollot. Structure-preserving manipulation of photographs. In *NPAR '07: Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 103–110, New York, NY, USA, 2007. ACM.
- [71] Sylvain Paris and Frédo Durand. A fast approximation of the bilateral filter using a signal processing approach. *Int. J. Comput. Vision*, 81(1):24–52, 2009.
- [72] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2003.
- [73] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7):629–639, Jul 1990.
- [74] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.*, 23(3):664–672, 2004.
- [75] Fatih Porikli. Constant time $O(1)$ bilateral filtering. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2008.
- [76] David M. Regan. *Human Perception of Objects: Early Visual Processing of Spatial Form Defined by Luminance, Color, Texture, Motion, and Binocular Disparity*. Sinauer Associates, 2000.
- [77] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital tapestry [automatic image synthesis]. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 589–596 vol. 1, June 2005.
- [78] Carsten Rother, Lucas Bordeaux, Youssef Hamadi, and Andrew Blake. Autocollage. *ACM Trans. Graph.*, 25(3):847–852, 2006.
- [79] Carsten Rother, Lucas Bordeaux, Youssef Hamadi, and Andrew Blake. Microsoft Autocollage 2008. Software, 2008.
- [80] David Salomon. *Data Compression: The Complete Reference*, chapter Statistical Methods, page 47. Springer, 2004.
- [81] Otmar Scherzer and Joachim Weickert. Relations between regularization and diffusion filtering. *J. Math. Imaging Vis.*, 12(1):43–63, 2000.

- [82] Stefan Schlechtweg, Tobias Germer, and Thomas Strothotte. Renderbots: Multi-agent systems for direct image generation. *Computer Graphics Forum*, 24(2):137–148, 2005.
- [83] Philippe G. Schyns and Aude Oliva. From blobs to boundary edges: Evidence for time- and spatial-scale dependent scene recognition. *Psychological Science*, 5(4):195–200, 1994.
- [84] Al Seckel. *Masters of Deception: Escher, Dali & Artists of Optical Illusion*. Sterling Publishing Co., Inc., 387 Park Avenue South, New York, NY 10016, 2004.
- [85] Michio Shiraishi and Yasushi Yamaguchi. An algorithm for automatic painterly rendering based on local source image approximation. In *NPAC '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 53–58, New York, NY, USA, 2000. ACM.
- [86] R. Silvers and M. Hawley. *Photomosaics*. Henry Holt and Co., 1997.
- [87] Jack Sklansky. Image segmentation and feature extraction. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(4):237–247, April 1978.
- [88] Sara L. Su, Frédo Durand, and Maneesh Agrawala. De-emphasis of distracting image regions using texture power maps. In *APGV '05: Proceedings of the 2nd symposium on Applied perception in graphics and visualization*, pages 164–164, New York, NY, USA, 2005. ACM.
- [89] Terrie Sultan. *Chuck Close Prints: Process and Collaboration*. Princeton University Press, 2003.
- [90] William B. Thompson. *Fundamentals of Computer Graphics*, chapter Chapter 21: Visual Perception, pages 477–519. A K Peters, Ltd., 888 Worcester Street, Suite 230, Wellesley, MA 02482, 2 edition, 2005.
- [91] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, Jan 1998.
- [92] Nicholas Tran. Generating photomosaics: an empirical study. In *SAC '99: Proceedings of the 1999 ACM symposium on Applied computing*, pages 105–109, New York, NY, USA, 1999. ACM.
- [93] Mihran Tuceryan and Anil K. Jain. *Handbook of Pattern Recognition and Computer Vision*, chapter Chapter 2.1: Texture Analysis, pages 207–248. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2 edition, 1998.
- [94] Robert Ulichney. *Digital Halftoning*. The MIT Press, Cambridge, Massachusetts, 1987.
- [95] Li-Yi Wei. Deterministic texture analysis and synthesis using tree structure vector quantization. In *SIBGRAPI '99: Proceedings of the XII Brazilian Symposium on Computer Graphics and Image Processing*, pages 207–214, Washington, DC, USA, 1999. IEEE Computer Society.
- [96] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [97] Holger Winnemöller, Sven C. Olsen, and Bruce Gooch. Real-time video abstraction. *ACM Trans. Graph.*, 25(3):1221–1226, 2006.
- [98] R. Y. Wong and E. L. Hall. Sequential hierarchical scene matching. *IEEE Trans. Comput.*, 27(4):359–366, 1978.
- [99] Yue Zhang. On the use of CBIR in image mosaic generation. Technical Report TR 02-17, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, 2002.

- [100] Q. Zheng and R. Chellappa. A computational vision approach to image registration. In *Pattern Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on*, pages 193–197, August, September 1992.
- [101] Barbara Zitová and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977 – 1000, 2003.